ELSEVIER

Available online at www.sciencedirect.com

ScienceDirect

Discrete Applied Mathematics 155 (2007) 2242–2256

DISCRETE APPLIED MATHEMATICS

www.elsevier.com/locate/dam

Finding a minimum path cover of a distance-hereditary graph in polynomial time[☆]

Ruo-Wei Hung[a,*], Maw-Shang Chang[b]

[a]Department of Computer Science and Information Engineering, Chaoyang University of Technology, Wufong, Taichung 413, Taiwan

[b]Department of Computer Science and Information Engineering, National Chung Cheng University, Ming-Hsiung, Chiayi 621, Taiwan

Received 23 January 2006; received in revised form 5 May 2007; accepted 3 June 2007

Available online 14 June 2007

Abstract

A *path cover* of a graph $G = (V, E)$ is a set of pairwise vertex-disjoint paths such that the disjoint union of the vertices of these paths equals the vertex set V of G. The *path cover problem* is, given a graph, to find a path cover having the minimum number of paths. The path cover problem contains the Hamiltonian path problem as a special case since finding a path cover, consisting of a single path, corresponds directly to the Hamiltonian path problem. A graph is a distance-hereditary graph if each pair of vertices is equidistant in every connected induced subgraph containing them. The complexity of the path cover problem on distance-hereditary graphs has remained unknown. In this paper, we propose the first polynomial-time algorithm, which runs in $O(|V|^9)$ time, to solve the path cover problem on distance-hereditary graphs.

Keywords: Graph algorithms; Hamiltonian path; Path cover; Distance-hereditary graphs

1. Introduction

All graphs considered in this paper are finite and undirected, without loops or multiple edges. Let $G = (V, E)$ be a graph with vertex set V and edge set E. Throughout this paper, let m and n denote the numbers of edges and vertices of G, respectively. A *Hamiltonian path* (resp. *Hamiltonian cycle*) in a graph is a simple path (resp. cycle) in which each vertex of the graph appears exactly once. A graph is called *Hamiltonian* if it contains a Hamiltonian cycle. The *Hamiltonian path problem* (resp. *Hamiltonian cycle problem*) is to determine whether a Hamiltonian path (resp. cycle) exists in a graph, and find one if such a path (resp. cycle) does exist. A *path cover* of a graph $G = (V, E)$ is a collection of vertex-disjoint paths $P_1 = (V_1, E_1)$, $P_2 = (V_2, E_2), \ldots, P_r = (V_r, E_r)$ in G whose union is V, where V_k and E_k are the vertex and edge sets of path P_k, respectively, i.e., $V_i \cap V_j = \emptyset$ for $i \neq j$ and $\bigcup_{i=1}^{r} V_i = V$. The *path cover number* of G, denoted by $\pi(G)$, is the smallest cardinality of a path cover of G. The *minimum path cover* of G is a path cover of G of size $\pi(G)$. The *path cover problem* is to find a minimum path cover of a graph. The path cover problem has received some alternative names in the literature, such as *optimal path cover* [1,31,33] and *path partition* [34].

☆ This research was partly supported by the National Science Council of Taiwan, ROC under Grant no. NSC93-2213-E252-007.

* Corresponding author. Tel.: +886 4 2332 3000x7758; fax: +886 4 2374 2375.

E-mail addresses: rwhung@cyut.edu.tw (R.-W. Hung), mschang@cs.ccu.edu.tw (M.-S. Chang).

0166-218X/$ - see front matter © 2007 Elsevier B.V. All rights reserved.

doi:10.1016/j.dam.2007.06.001

It is evident that the path cover problem for general graphs is NP-complete since finding a path cover, consisting of a single path, corresponds directly to the Hamiltonian path problem [16]. Polynomial-time algorithms have been known for only a few special classes of graphs, including trees [27], block graphs [33,34], interval graphs [1], circular-arc graphs [24], cographs [26], bipartite permutation graphs [31] and cocomparability graphs [11]. The path cover problem has many practical applications in different areas, including establishing ring protocol [32], mapping parallel programs to parallel architectures [27,30], code optimization [3] and program testing [29].

A connected graph is *distance-hereditary* if the distance between every two vertices in any connected induced subgraph is the same as in the original graph, where the *distance* between two vertices is the length of a shortest path connecting them. Distance-hereditary graphs form a subclass of perfect graphs that are graphs G in which the maximum clique size equals the chromatic number for every induced subgraph of G [4,17]. Examples of graphs belonging to the class of distance-hereditary graphs are trees, complete graphs, k-partite complete graphs, block graphs, cographs, Ptolemaic graphs and bipartite distance-hereditary graphs [4,6,7,21].

Distance-hereditary graphs have been exploited in the design of interconnection network topologies [13,14]. Esfahanian and Oellermann modeled a distance-hereditary graph as a computer network DhN [14]. The path cover problem on DhN has many practical applications. For example, in order to establish ring protocol on DhN [32], the computer network DhN may be augmented by some auxiliary edges so as to make it Hamiltonian [19]. It is easily verified that the minimum number of additional edges needed to make DhN Hamiltonian is identical to the path cover number of DhN. Other notable application of the path cover problem on distance-hereditary graphs includes mapping parallel programs into DhN [27,30].

Several properties of distance-hereditary graphs were also explored for algorithmic applications. Bandelt and Mulder showed that the graphs of house, domino, gem and hole are neither induced subgraphs nor isometric subgraphs of a distance-hereditary graph [2]. They also showed that a distance-hereditary graph can be constructed from an isolated vertex by adding vertices one by one through operations called *one-vertex extensions* [2]. Hammer and Maffray proposed a linear-time recognition algorithm that constructs a sequence of one-vertex extensions for a distance-hereditary graph [20]. Damiand et al. pointed out that the recognition algorithm in [20] was slightly incorrect and gave a correct linear-time recognition algorithm for distance-hereditary graphs [12]. Chang et al. gave a new recursive definition for distance-hereditary graphs and showed that the weighted vertex cover problem, the weighted independent domination problem, the minimum fill-in problem and the treewidth problem on distance-hereditary graphs are polynomially solvable [5].

Golumbic and Rotics showed that the clique-width of every distance-hereditary graph is at most three [18]. The notion of clique-width of graphs was first introduced by Courcelle et al. [8]. The *clique-width* of a graph G is defined as the minimum number of labels needed to construct G, using four graph operations: creation of a new vertex with label i, disjoint union, connecting vertices with specified labels and renaming labels. An expression built from the above four operations using k labels is called a *k-expression*. Each k-expression uniquely defines a graph. For more background on clique-width, we refer the reader to [8,10]. A graph class $\mathscr{C}$ is of *bounded clique-width* if for every graph G in $\mathscr{C}$ there is a fixed integer k such that the clique-width of G is not greater than k. Bounded clique-width graphs are especially interesting from algorithmic point of view. A lot of NP-complete problems can be solved in polynomial time for graphs of bounded clique-width if an expression for the input graph is explicitly given. A graph problem on bounded clique-width graphs is said to be an MS_1 problem if it can be defined by a monadic second order logic (MS-logic) formula, using quantifiers on vertices but not on edges. A graph problem is called an MS_2 problem if it is definable in MS-logic formula with quantifiers on both vertices and edges. Courcelle et al. showed an elegant result that all MS_1 problems on bounded clique-width graphs can be solved in linear time if an expression for the input graph is explicitly given [9]. Golumbic and Rotics showed that a corresponding 3-expression for a distance-hereditary graph can be built in linear time [18]. Therefore, a wide class of graph problems are linear-time solvable on distance-hereditary graphs. Note that the Hamiltonian and path cover problems are not MS_1 problems since they cannot be represented by MS-logic formula using quantifiers over vertex set only. The technique in [9,18] cannot be applied to solve the problem considered in this paper. However, Espelage et al. proposed polynomial-time algorithms to solve some problems which are not MS_1 problems on bounded clique-width graphs [15]. They solved the Hamiltonian path problem for graphs with bounded clique-width k in $\mathrm{O}(n^{k^2})$ time. That is, the algorithm proposed by Espelage et al. for the Hamiltonian path problem on distance-hereditary graphs runs in $\mathrm{O}(n^9)$ time [15]. The path cover problem on bounded clique-width graphs is still open.

Whether the path cover problem on distance-hereditary graphs can be solved in polynomial time has remained unknown. In this paper, we present the first polynomial-time algorithm, which uses the dynamic programming method and runs in $O(n^9)$ time, to solve the path cover problem on distance-hereditary graphs. Previous related works are summarized below. Nicolai presented the first polynomial-time algorithms, which run in $O(n^3)$ and $O(n^5)$ times, respectively, for the Hamiltonian cycle and path problems, respectively, on distance-hereditary graphs [28]. We solved the Hamiltonian cycle problem on distance-hereditary graphs in $O(n^2)$ time [25]. Hsieh et al. improved the above result to obtain an $O(m+n)$-linear-time algorithm for the Hamiltonian cycle problem on distance-hereditary graphs [22]. Recently, we present a unified approach to solving the Hamiltonian cycle and Hamiltonian path problems on distance-hereditary graphs in $O(m+n)$ linear time [23].

The rest of this paper is organized as follows. In Section 2, we review some properties of distance-hereditary graphs and give some basic definitions. Section 3 proves some lemmas and theorems which are used in designing our polynomial-time algorithm. Finally, we present a dynamic programming polynomial-time algorithm to solve the path cover problem on distance-hereditary graphs in Section 4.

2. Preliminaries

Chang et al. have shown that distance-hereditary graphs have an elegant characterization [5]. The characterization makes use of the concept of twin sets. Every distance-hereditary graph has a *twin set* that is a subset of vertices. We use $TS(G)$ to denote a twin set of a distance-hereditary graph G in the following.

Definition 2.1 (*Chang et al. [5]*). The class of distance-hereditary graphs can be defined by the following recursive definition:

(1) A graph consisting of a single vertex v is a distance-hereditary graph with the twin set $\{v\}$.
(2) If G_L and G_R are distance-hereditary graphs, then the union G of G_L and G_R is a distance-hereditary graph and $TS(G)=TS(G_L)\cup TS(G_R)$. In this case, we say that G is formed from G_L and G_R by a *false-twin operation*.
(3) If G_L and G_R are distance-hereditary graphs, then the graph G obtained from G_L and G_R by connecting every vertex of $TS(G_L)$ to all vertices of $TS(G_R)$ is a distance-hereditary graph and $TS(G)=TS(G_L)\cup TS(G_R)$. In this case, we say that G is formed from G_L and G_R by a *true-twin operation*.
(4) If G_L and G_R are distance-hereditary graphs, then the graph G obtained from G_L and G_R by connecting every vertex of $TS(G_L)$ to all vertices of $TS(G_R)$ is a distance-hereditary graph and $TS(G)=TS(G_L)$. In this case, we say that G is formed from G_L and G_R by a *pendant operation*.

A vertex of G is called a *twin vertex* if it is in $TS(G)$, and is called a *non-twin vertex* otherwise.

Notice that the two graphs formed from G_L and G_R by a true-twin operation and by a pendant operation are isomorphic but with different twin sets. In the rest of the paper, we assume the twin set of G is the twin set of G_L whenever we say that G is formed from G_L and G_R by a pendant operation.

Using the above definition, a binary ordered decomposition tree $DT(G)$ of a distance-hereditary graph G can be constructed. The decomposition tree of a distance-hereditary graph is a binary rooted node-labeled tree and is defined as follows:

Definition 2.2 (*Chang et al. [5]*). The decomposition tree $DT(G)$ of a distance-hereditary graph G consisting of a single vertex v is a tree of one node labeled by v. If G is formed from G_L and G_R by a false-twin (resp. true-twin, pendant) operation, then the root of the decomposition tree $DT(G)$ is a node labeled by 'F' (resp. 'T', 'P') with the roots of $DT(G_L)$ and $DT(G_R)$ being the left and right children of the root of $DT(G)$, respectively.

Notice that the decomposition tree $DT(G)$ is a binary ordered tree. For instance, given the distance-hereditary graph G shown in Fig. 1(a), the decomposition tree $DT(G)$ of G is shown in Fig. 1(b). Note that if G is formed from G_L and G_R by a pendant operation, then $DT(G_L)$ and $DT(G_R)$ are the left and right subtrees of $DT(G)$, respectively, and $TS(G)=TS(G_L)$.

Theorem 2.1 (*Chang et al. [5]*). *A decomposition tree of a distance-hereditary graph can be constructed in* $O(m+n)$ *time*.

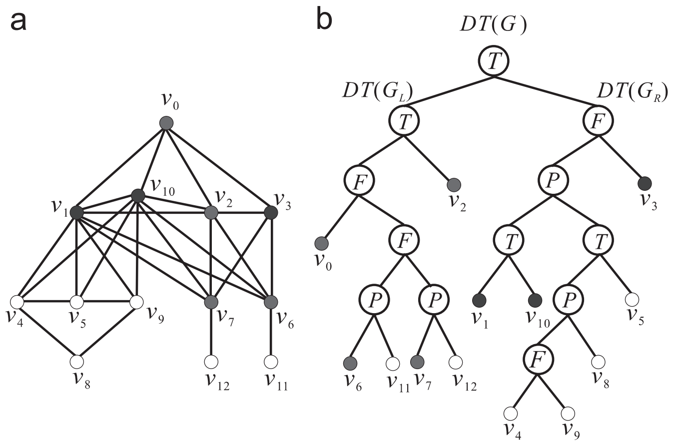

Fig. 1. (a) A distance-hereditary graph G and (b) a decomposition tree $DT(G)$ of G, where the vertices in $TS(G)$ are drawn by filled circles.

In the remainder of the paper, we assume that $G=(V,E)$ is a distance-hereditary graph and is formed from G_L and G_R by one of a false-twin operation, a true-twin operation or a pendant operation. We use V_L and V_R to denote the vertex sets of G_L and G_R, respectively. In other words, $V=V_L\cup V_R$ and $V_L\cap V_R=\emptyset$. For any two sets X and Y, let $X\backslash Y$ denote the set of elements of X that are not in Y.

Notice that every vertex in $TS(G_L)$ is adjacent to all vertices of $TS(G_R)$, and is not adjacent to any vertex in $V_R\backslash TS(G_R)$, if G is formed from G_L and G_R by either a true-twin operation or a pendant operation. By symmetry, every vertex in $TS(G_R)$ is adjacent to all vertices of $TS(G_L)$, and is not adjacent to any vertex in $V_L\backslash TS(G_L)$, if G is formed from G_L and G_R by either a true-twin operation or a pendant operation.

Definition 2.3. A path P, denoted by $v_1v_2\cdots v_{|P|}$, is a sequence $(v_1,v_2,\ldots,v_{|P|})$ of vertices, each appearing exactly once, on which v_i and v_{i+1} are adjacent for $1\leqslant i\leqslant |P|-1$. The first and last vertices visited by path P are denoted by *start*(P) and *end*(P), respectively. Both of them are *end vertices* of P. We allow $start(P)$ and *end*(P) to be the same only in the case that P contains exactly one vertex.

Definition 2.4. Let $P_1=x_1x_2\cdots x_{|P_1|}$ and $P_2=y_1y_2\cdots y_{|P_2|}$ be two vertex-disjoint paths of a graph such that $x_{|P_1|}$ and y_1 are adjacent in the graph. The *coalescence* of P_1 and P_2, denoted by $P_1\dot{+}P_2$, is defined as $x_1x_2\cdots x_{|P_1|}y_1y_2\cdots y_{|P_2|}$.

Definition 2.5. Let G be a distance-hereditary graph. A path of G is called a *twin path* if both of its two end vertices are in $TS(G)$. A path of G is called a *semi-twin path* if exactly one of its end vertices is in $TS(G)$. A path of G is called a *non-twin path* if neither of its two end vertices is in $TS(G)$.

By the above definition, a twin path of G may consist of only one vertex in $TS(G)$, a non-twin path of G may consist of only one vertex not in $TS(G)$ and a semi-twin path of G must contain at least two vertices with one in $TS(G)$ and the other not in $TS(G)$.

Definition 2.6. A path cover $\mathscr{PC}$ of a distance-hereditary graph G is called a (t,s,u)-*constrained path cover* if (1) $|\mathscr{PC}|=t+s+u$; (2) there are exactly t twin paths in $\mathscr{PC}$; (3) there are exactly s semi-twin paths in $\mathscr{PC}$ and (4) all other paths in $\mathscr{PC}$ are non-twin paths.

A path of a distance-hereditary graph G is clearly either a twin path, a semi-twin path or a non-twin path of G. Hence, the following proposition immediately holds:

Proposition 2.2. *Assume that G is a distance-hereditary graph. Then, $\mathscr{PC}$ is a path cover of G if and only if it is a constrained path cover of G.*

Definition 2.7. Let G be a distance-hereditary graph. Define $\mathscr{F}(G) = \{(t, s, u) | G$ has a (t, s, u)-constrained path cover$\}$.

By Proposition 2.2, $\pi(G) = \min\{t + s + u | (t, s, u) \in \mathscr{F}(G)\}$ for a distance-hereditary graph G.

Definition 2.8. Let $\mathscr{P}C$ be a constrained path cover of a distance-hereditary graph G. Define $T(G, \mathscr{P}C)$, $S(G, \mathscr{P}C)$ and $U(G, \mathscr{P}C)$ to be the subsets of $\mathscr{P}C$ consisting of all twin paths, all semi-twin paths and all non-twin paths in $\mathscr{P}C$, respectively.

Definition 2.9. Let W be a subset of vertices of G and let P be a path of G. A subpath P' of path P is called a W-subpath of P if P' visits vertices in W only. A W-subpath of P is W-*maximal* if it is not a proper subpath of any W-subpath of P. For a set $\mathscr{P}$ of vertex-disjoint paths of G, denote by $W(\mathscr{P})$ the set of all W-maximal subpaths of all paths in $\mathscr{P}$.

Let $\mathscr{P}C$ be a (t, s, u)-constrained path cover of a distance-hereditary graph G. Then, $V_L(\mathscr{P}C)$ and $V_R(\mathscr{P}C)$ are the sets of all V_L-maximal and V_R-maximal subpaths of all paths in $\mathscr{P}C$, respectively. $V_L(\mathscr{P}C)$ and $V_R(\mathscr{P}C)$ are clearly constrained path covers of G_L and G_R, respectively. On the other hand, a V_L-maximal subpath is called a V_L-maximal twin subpath (resp. semi-twin subpath, non-twin subpath) if it is a twin path (resp. semi-twin path, non-twin path) of G_L. By symmetry, a V_R-maximal subpath is similarly defined. By definition, $\mathscr{P}C = T(G, \mathscr{P}C) \cup S(G, \mathscr{P}C) \cup U(G, \mathscr{P}C)$ and each one of $T(G, \mathscr{P}C)$, $S(G, \mathscr{P}C)$ and $U(G, \mathscr{P}C)$ consists of a set of vertex-disjoint paths. We can see that $V_L(T(G, \mathscr{P}C))$, $V_L(S(G, \mathscr{P}C))$ and $V_L(U(G, \mathscr{P}C))$ contain all V_L-maximal subpaths of paths in $T(G, \mathscr{P}C)$, $S(G, \mathscr{P}C)$ and $U(G, \mathscr{P}C)$, respectively. $V_R(T(G, \mathscr{P}C))$, $V_R(S(G, \mathscr{P}C))$ and $V_R(U(G, \mathscr{P}C))$ are similarly defined.

We give an example to illustrate the above definitions. Consider the distance-hereditary graph $G = (V, E)$ and its decomposition tree $DT(G)$ shown in Fig. 1. Then, G_L and G_R are the subgraphs of G induced by the leaves of $DT(G_L)$ and $DT(G_R)$, respectively, and V_L, V_R, $TS(G_L)$ and $TS(G_R)$ are shown in Fig. 2, where $V_L \backslash TS(G_L) = \{v_{11}, v_{12}\}$. Let $P_t = v_3v_0v_2$, $P_{s_1} = v_{11}v_6$, $P_{s_2} = v_{12}v_7$, $P_u = v_8v_4v_1v_{10}v_5v_9$ and let $\mathscr{P}C = \{P_t, P_{s_1}, P_{s_2}, P_u\}$. Then, $\mathscr{P}C$ is a $(1, 2, 1)$-constrained path cover of G. We can see from definition that $T(G, \mathscr{P}C) = \{P_t\}$, $S(G, \mathscr{P}C) = \{P_{s_1}, P_{s_2}\}$, $U(G, \mathscr{P}C) = \{P_u\}$, $V_L(\mathscr{P}C) = \{P_{s_1}, P_{s_2}, v_0v_2\}$ and $V_R(\mathscr{P}C) = \{P_u, v_3\}$. Then, $T(G_L, V_L(\mathscr{P}C)) = V_L(T(G, \mathscr{P}C)) = \{v_0v_2\}$, $S(G_L, V_L(\mathscr{P}C)) = V_L(S(G, \mathscr{P}C)) = \{P_{s_1}, P_{s_2}\}$, $U(G_L, V_L(\mathscr{P}C)) = V_L(U(G, \mathscr{P}C)) = \emptyset$, $T(G_R, V_R(\mathscr{P}C)) = V_R(T(G, \mathscr{P}C)) = \{v_3\}$, $S(G_R, V_R(\mathscr{P}C)) = V_R(S(G, \mathscr{P}C)) = \emptyset$ and $U(G_R, V_R(\mathscr{P}C)) = V_R(U(G, \mathscr{P}C)) = \{P_u\}$.

We next define three operations on two distinct sets of vertex-disjoint paths as follows:

Definition 2.10. ($\lhd$ operation) Let $\mathscr{P} = \{P_1, P_2, \ldots, P_p\}$ and $\mathscr{Q} = \{Q_1, Q_2, \ldots, Q_q\}$ be two distinct sets of vertex-disjoint paths such that $p \geqslant q$ and *end*(P_i) is adjacent to *start*(Q_i) for $i \leqslant q$. For an integer $k \leqslant q$, define $\lhd_{[k]}$ to be the operation on $\mathscr{P}$ and $\mathscr{Q}$, denoted by $\mathscr{P} \lhd_{[k]} \mathscr{Q}$, that constructs a set of vertex-disjoint paths $\{P_1 \dot{+} Q_1, P_2 \dot{+} Q_2, \ldots, P_k \dot{+} Q_k, P_{k+1}, \ldots, P_p, Q_{k+1}, \ldots, Q_q\}$. Fig. 3(a) depicts the operation.

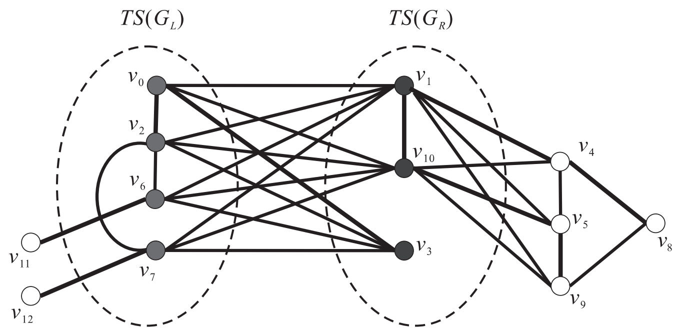

Fig. 2. An illustration of $TS(G_L)$ and $TS(G_R)$ of the distance-hereditary graph G shown in Fig. 1(a), where the vertices in $TS(G_L) \cup TS(G_R)$ are drawn by filled circles.

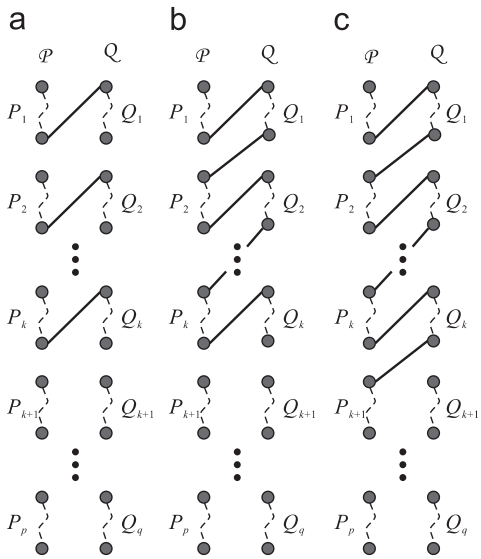

Fig. 3. Operations defined on two distinct sets $\mathscr{P}$ and $\mathscr{Q}$ of vertex-disjoint paths, where (a) $\mathscr{P}\lhd_{[k]}\mathscr{Q}$, (b) $\mathscr{P}\bowtie_{[k]}\mathscr{Q}$ and (c) $\mathscr{P}\odot_{[k]}\mathscr{Q}$.

Definition 2.11. ($\bowtie$ operation) Let $\mathscr{P}=\{P_1, P_2, \ldots, P_p\}$ and $\mathscr{Q}=\{Q_1, Q_2, \ldots, Q_q\}$ be two distinct sets of vertex-disjoint paths such that both end vertices of every path in $\mathscr{P}$ are adjacent to all end vertices of paths in $\mathscr{Q}$ and $p\geqslant q$. For an integer $k\leqslant q$, define $\bowtie_{[k]}$ to be the operation on $\mathscr{P}$ and $\mathscr{Q}$, denoted by $\mathscr{P}\bowtie_{[k]}\mathscr{Q}$, that generates a set of vertex-disjoint paths $\{P_1\dotplus Q_1\dotplus P_2\dotplus Q_2\dotplus\cdots\dotplus P_k\dotplus Q_k, P_{k+1}, \ldots, P_p, Q_{k+1}, \ldots, Q_q\}$. The operation is depicted in Fig. 3(b).

Definition 2.12. ($\odot$ operation) Let $\mathscr{P}=\{P_1, P_2, \ldots, P_p\}$ and $\mathscr{Q}=\{Q_1, Q_2, \ldots, Q_q\}$ be two distinct sets of vertex-disjoint paths such that both end vertices of every path in $\mathscr{P}$ are adjacent to all end vertices of paths in $\mathscr{Q}$ and $p>q$. For an integer $k\leqslant q$, define $\odot_{[k]}$ to be the operation on $\mathscr{P}$ and $\mathscr{Q}$, denoted by $\mathscr{P}\odot_{[k]}\mathscr{Q}$, that generates a set of vertex-disjoint paths $\{P_1\dotplus Q_1\dotplus P_2\dotplus Q_2\dotplus\cdots\dotplus P_k\dotplus Q_k\dotplus P_{k+1}, P_{k+2}, \ldots, P_p, Q_{k+1}, \ldots, Q_q\}$. Fig. 3(c) reveals the operation.

3. The path cover problem on distance-hereditary graphs

In this section, we will prove some lemmas and theorems which are used in designing our dynamic programming algorithm. The following two lemmas can be easily verified:

Lemma 3.1. *Assume G is a distance-hereditary graph consisting of a single vertex. Then,* $\mathscr{F}(G)=\{(1,0,0)\}$.

Lemma 3.2. *Assume G is a distance-hereditary graph formed from G_L and G_R by a false-twin operation. Then,* $\mathscr{F}(G)=\{(t_L+t_R, s_L+s_R, u_L+u_R)|(t_L, s_L, u_L)\in\mathscr{F}(G_L)$ *and* $(t_R, s_R, u_R)\in\mathscr{F}(G_R)\}$.

In the rest of this section, we will show necessary and sufficient conditions for a distance-hereditary graph G to have a constrained path cover when G is formed from G_L and G_R by either a true-twin operation or a pendant operation.

Lemma 3.3. *Assume G is a distance-hereditary graph formed from G_L and G_R by a true-twin operation and G has a (t, s, u)-constrained path cover $\mathscr{P}C$. Then, $V_L(\mathscr{P}C)$ and $V_R(\mathscr{P}C)$ are (t_L, s_L, u_L)-constrained and (t_R, s_R, u_R)-constrained path covers of G_L and G_R, respectively, satisfying the following conditions*:

(1) $\bar{u}=u-u_L-u_R\geqslant 0$;
(2) $s+2u=(s_L+s_R)+2(u_L+u_R)$;
(3) *if* $t=s=\bar{u}=0$, *then* $t_L=t_R=0$; *otherwise*, $\max\{0, t_L-t_R-s_R+\bar{u}, t_R-t_L-s_L+\bar{u}\}\leqslant t\leqslant t_L+t_R-\max\{0, \bar{u}-s_L, \bar{u}-s_R\}$.

Proof. By definition, every path in $U(G_L, V_L(\mathscr{P}C))$ and $U(G_R, V_R(\mathscr{P}C))$ only visits vertices in V_L and V_R, respectively, and is a non-twin path of G in $\mathscr{P}C$. Hence, $u \geqslant u_L + u_R$ and Condition (1) is satisfied.

$s + 2u$ is the number of non-twin vertices of G that are end vertices of paths in $\mathscr{P}C$. Similarly, $s_L + 2u_L + s_R + 2u_R$ is the number of non-twin vertices of G_L together with those of G_R that are end vertices of subpaths in $\mathscr{P}C$. It is easy to justify that a non-twin vertex of G is an end vertex of a path in $\mathscr{P}C$ if and only if it is an end vertex of a path in $V_L(\mathscr{P}C)$ or $V_R(\mathscr{P}C)$. Therefore, $s + 2u = s_L + 2u_L + s_R + 2u_R$ and Condition (2) is satisfied.

We next prove that Condition (3) is satisfied. Let t_{L_1}, t_{L_2} and t_{L_3} be the numbers of twin paths of G_L in $V_L(T(G, \mathscr{P}C))$, $V_L(S(G, \mathscr{P}C))$ and $V_L(U(G, \mathscr{P}C))$, respectively. Similarly, let t_{R_1}, t_{R_2} and t_{R_3} be the numbers of twin paths of G_R in $V_R(T(G, \mathscr{P}C))$, $V_R(S(G, \mathscr{P}C))$ and $V_R(U(G, \mathscr{P}C))$, respectively. Then, we have that

$$t_L = t_{L_1} + t_{L_2} + t_{L_3}, \tag{1}$$

$$t_R = t_{R_1} + t_{R_2} + t_{R_3}. \tag{2}$$

We can easily see that $t \leqslant t_{L_1} + t_{R_1}$. Suppose $t = s = \bar{u} = 0$. Then, $t_{L_1} = t_{R_1} = t_{L_2} = t_{R_2} = 0$ since $t = s = 0$. Since $\bar{u} = 0$, there exists no twin path of G_L and G_R in $V_L(U(G, \mathscr{P}C))$ and $V_R(U(G, \mathscr{P}C))$, respectively, and, hence, $t_{L_3} = t_{R_3} = 0$. By Eqs. (1) and (2), we get that $t_L = t_R = 0$. In the following, suppose that $t \neq 0$, $s \neq 0$ or $\bar{u} \neq 0$. Let s_{L_1} and s_{L_2} be the numbers of semi-twin paths of G_L in $V_L(S(G, \mathscr{P}C))$ and $V_L(U(G, \mathscr{P}C))$, respectively. Similarly, let s_{R_1} and s_{R_2} be the numbers of semi-twin paths of G_R in $V_R(S(G, \mathscr{P}C))$ and $V_R(U(G, \mathscr{P}C))$, respectively. Since exactly two paths of $S(G_L, V_L(\mathscr{P}C)) \cup S(G_R, V_R(\mathscr{P}C))$ are used to form a path of $U(G, \mathscr{P}C) \setminus (U(G_L, V_L(\mathscr{P}C)) \cup U(G_R, V_R(\mathscr{P}C)))$, we have that

$$\bar{u} = u - u_L - u_R = \frac{s_{L_2} + s_{R_2}}{2}. \tag{3}$$

By definition, both end vertices of every path in $T(G, \mathscr{P}C)$ are twin vertices of G, where $TS(G) = TS(G_L) \cup TS(G_R)$. There are three types of paths in $T(G, \mathscr{P}C)$: type-1 paths are those paths with one end vertex in $TS(G_L)$ and the other end vertex in $TS(G_R)$; type-2 paths are those paths with both end vertices in $TS(G_L)$ and type-3 paths are those paths with both end vertices in $TS(G_R)$. Let t_1, t_2 and t_3 be the numbers of the paths of type-1, type-2 and type-3, respectively. Then, $|T(G, \mathscr{P}C)| = t = t_1 + t_2 + t_3$. Every type-1 path has as many V_L-maximal twin subpaths as V_R-maximal twin subpaths. Every type-2 path has exactly one more V_L-maximal twin subpath than V_R-maximal twin subpaths. Every type-3 path has exactly one more V_R-maximal twin subpath than V_L-maximal twin subpaths. Thus, $t_{L_1} - t_{R_1} = t_2 - t_3$. Hence, $t - (t_{L_1} - t_{R_1}) = t_1 + 2t_3 \geqslant 0$, i.e., $t \geqslant t_{L_1} - t_{R_1}$. By symmetry, $t - (t_{R_1} - t_{L_1}) = t_1 + 2t_2 \geqslant 0$, i.e., $t_{L_1} - t_{R_1} \geqslant -t$. Combining the above two equations, we get

$$-t \leqslant t_{L_1} - t_{R_1} \leqslant t. \tag{4}$$

By definition, one end vertex of every path in $S(G, \mathscr{P}C)$ is a non-twin vertex of G, while the other end vertex is a twin vertex of G. There are four types of paths in $S(G, \mathscr{P}C)$: type-I paths are those paths with both end vertices in V_L; type-II paths are those paths with both end vertices in V_R; type-III paths are those paths with one end vertex in $V_L \setminus TS(G_L)$ and the other end vertex in $TS(G_R)$ and type-IV paths are those paths with one end vertex in $V_R \setminus TS(G_R)$ and the other end vertex in $TS(G_L)$. Let s_1, s_2, s_3 and s_4 be the numbers of the paths of type-I, type-II, type-III and type-IV, respectively. Then, $s_{L_1} = s_1 + s_3$ and $s_{R_1} = s_2 + s_4$. Every path of type-I and type-II has as many V_L-maximal twin subpaths as V_R-maximal twin subpaths. Every type-III path has exactly one more V_R-maximal twin subpath than V_L-maximal twin subpaths. Every type-IV path has exactly one more V_L-maximal twin subpath than V_R-maximal twin subpaths. Thus, $t_{L_2} - t_{R_2} = s_4 - s_3$. Since $0 \leqslant s_3 \leqslant s_{L_1}$ and $0 \leqslant s_4 \leqslant s_{R_1}$, we have that $-s_{L_1} \leqslant s_4 - s_3 \leqslant s_{R_1}$. Hence, we have

$$-s_{L_1} \leqslant t_{L_2} - t_{R_2} \leqslant s_{R_1}. \tag{5}$$

By definition, both end vertices of every path in $U(G, \mathscr{P}C)$ are non-twin vertices of G. There are five types of paths in $U(G, \mathscr{P}C)$: type-A paths are those paths visiting vertices in V_L only; type-B paths are those paths visiting vertices in V_R only; type-C paths are those paths visiting vertices in both V_L and V_R with both end vertices in $V_L \setminus TS(G_L)$; type-D paths are those paths visiting vertices in both V_L and V_R with both end vertices in $V_R \setminus TS(G_R)$ and type-E paths are those paths visiting vertices in both V_L and V_R with one end vertex in $V_L \setminus TS(G_L)$ and the other end vertex in $V_R \setminus S(G_R)$. Let u_1, u_2, u_3, u_4 and u_5 be the numbers of the paths of type-A, type-B, type-C, type-D and type-E, respectively. Then, $u_1 = u_L$, $u_2 = u_R$, $s_{L_2} = 2u_3 + u_5$, $s_{R_2} = 2u_4 + u_5$, $u = u_1 + u_2 + u_3 + u_4 + u_5$ and $\bar{u} = u_3 + u_4 + u_5$. Every type-A path is also a non-twin path of G_L and every type-B path is also a non-twin path of G_R. Every type-E path has as many V_L-maximal twin subpaths as V_R-maximal twin subpaths. Every type-C path has exactly one more V_R-maximal twin subpath than V_L-maximal twin subpaths. Every type-D path has exactly onemore V_L-maximal twin subpath than V_R-maximal twin subpaths. Thus, $t_{L_3} - t_{R_3} = u_4 - u_3$. Since $s_{L_2} = 2u_3 + u_5$ and $s_{R_2} = 2u_4 + u_5$, we have that $u_4 - u_3 = (s_{R_2} - s_{L_2})/2$. Hence, we get

$$t_{L_3} - t_{R_3} = \frac{s_{R_2} - s_{L_2}}{2}. \tag{6}$$

Combining Eqs. (1)–(2) and (4)–(6), we get that $-(t + s_{L_1} - (s_{R_2} - s_{L_2})/2) \leqslant t_L - t_R \leqslant t + s_{R_1} + (s_{R_2} - s_{L_2})/2$. By definition and Eq. (3), $\bar{u} = (s_{L_2} + s_{R_2})/2$, $s_L = s_{L_1} + s_{L_2}$ and $s_R = s_{R_1} + s_{R_2}$. Hence, $-(t + s_L - \bar{u}) \leqslant t_L - t_R \leqslant t + s_R - \bar{u}$. Then, $t \geqslant t_L - t_R - s_R + \bar{u}$ and $t \geqslant t_R - t_L - s_L + \bar{u}$. Obviously, $t \geqslant 0$. Hence, we have that

$$\max\{0, t_L - t_R - s_R + \bar{u}, t_R - t_L - s_L + \bar{u}\} \leqslant t. \tag{7}$$

By Eq. (6), we get that $t_{L_3} + t_{R_3} = 2t_{R_3} + (s_{R_2} - s_{L_2})/2 = 2t_{L_3} + (s_{L_2} - s_{R_2})/2$. By Eq. (3), $\bar{u} = (s_{L_2} + s_{R_2})/2$ and, hence, $(s_{R_2} - s_{L_2})/2 = \bar{u} - s_{L_2}$ and $(s_{L_2} - s_{R_2})/2 = \bar{u} - s_{R_2}$. Hence, $t_{L_3} + t_{R_3} = 2t_{R_3} + \bar{u} - s_{L_2} = 2t_{L_3} + \bar{u} - s_{R_2}$. Clearly, $t_{L_3} \geqslant 0$, $s_{L_2} \leqslant s_L$, $t_{R_3} \geqslant 0$ and $s_{R_2} \leqslant s_R$. Hence, $t_{L_3} + t_{R_3} \geqslant \bar{u} - s_L$ and $t_{L_3} + t_{R_3} \geqslant \bar{u} - s_R$. Obviously, $t_{L_3} + t_{R_3} \geqslant 0$. Hence, we get that $t_{L_3} + t_{R_3} \geqslant \max\{0, \bar{u} - s_L, \bar{u} - s_R\}$. It is easy to see that $t \leqslant t_{L_1} + t_{R_1} = (t_L - t_{L_2} - t_{L_3}) + (t_R - t_{R_2} - t_{R_3}) = t_L + t_R - (t_{L_2} + t_{R_2}) - (t_{L_3} + t_{R_3})$ and $t_{L_2} + t_{R_2} \geqslant 0$. Hence, we get that

$$t \leqslant t_L + t_R - \max\{0, \bar{u} - s_L, \bar{u} - s_R\}. \tag{8}$$

Combining Eqs. (7) and (8), we get that $\max\{0, t_L - t_R - s_R + \bar{u}, t_R - t_L - s_L + \bar{u}\} \leqslant t \leqslant t_L + t_R - \max\{0, \bar{u} - s_L, \bar{u} - s_R\}$. Thus, $\max\{0, t_L - t_R - s_R + \bar{u}, t_R - t_L - s_L + \bar{u}\} \leqslant t \leqslant t_L + t_R - \max\{0, \bar{u} - s_L, \bar{u} - s_R\}$ if $t \neq 0$, $s \neq 0$ or $\bar{u} \neq 0$. This completes the proof. $\square$

Lemma 3.4. *Assume G is a distance-hereditary graph formed from G_L and G_R by a true-twin operation. Let $\mathscr{P}_L$ and $\mathscr{P}_R$ be two sets of vertex-disjoint twin paths of G_L and G_R, respectively, such that $|\mathscr{P}_L| = t_L$, $|\mathscr{P}_R| = t_R$ and $t_L + t_R \geqslant 1$. Then, for any number t, where $t_L + t_R \geqslant t \geqslant \max\{1, t_L - t_R, t_R - t_L\}$, there exists a set $\mathscr{P}$ of vertex-disjoint twin paths of G such that $|\mathscr{P}| = t$, $V_L(\mathscr{P}) = \mathscr{P}_L$ and $V_R(\mathscr{P}) = \mathscr{P}_R$.*

Proof. For simplicity, let $h = \max\{1, t_L - t_R, t_R - t_L\}$. We will prove this lemma by showing that the following two statements are true:

(1) there exists a set $\mathscr{P}_h$ of vertex-disjoint twin paths of G such that $|\mathscr{P}_h| = h$, $V_L(\mathscr{P}_h) = \mathscr{P}_L$ and $V_R(\mathscr{P}_h) = \mathscr{P}_R$; and
(2) for any number t with $t_L + t_R \geqslant t \geqslant h$, we can obtain a set $\mathscr{P}$ of vertex-disjoint twin paths of G from $\mathscr{P}_h$ such that $|\mathscr{P}| = t$, $V_L(\mathscr{P}) = \mathscr{P}_L$ and $V_R(\mathscr{P}) = \mathscr{P}_R$.

We prove statement (1) first. By symmetry, assume $t_L - t_R \geqslant 0$. Consider the following two cases:

Case 1: $t_L = t_R$. Let $\mathscr{P}_h = \mathscr{P}_L \bowtie_{[t_R]} \mathscr{P}_R$. Then, $\mathscr{P}_h$ consists of only one twin path of G with one end vertex in $TS(G_L)$ and the other end vertex in $TS(G_R)$. Clearly, $h = 1$ in this case and $\mathscr{P}_h$ is a set consisting of a twin path of G with $V_L(\mathscr{P}_h) = \mathscr{P}_L$ and $V_R(\mathscr{P}_h) = \mathscr{P}_R$.

Case 2: $t_L > t_R$. In this case, $h = t_L - t_R$. Let $\mathscr{P}_h = \mathscr{P}_L$ if $t_R = 0$; otherwise, let $\mathscr{P}_h = \mathscr{P}_L \odot_{[t_R]} \mathscr{P}_R$. Then, $\mathscr{P}_h$ is a set of vertex-disjoint twin paths of G of size h such that $V_L(\mathscr{P}_h) = \mathscr{P}_L$ and $V_R(\mathscr{P}_h) = \mathscr{P}_R$. This proves statement (1).

In the above, we have shown that there exists a set $\mathscr{P}_h$ of vertex-disjoint twin paths of G such that $|\mathscr{P}_h| = h$, $V_L(\mathscr{P}_h) = \mathscr{P}_L$ and $V_R(\mathscr{P}_h) = \mathscr{P}_R$. For $|\mathscr{P}_L| + |\mathscr{P}_R| \geqslant t \geqslant h$, we can obtain a set $\mathscr{P}$ of vertex-disjoint twin paths of G from $\mathscr{P}_h$ such that $|\mathscr{P}| = t$, $V_L(\mathscr{P}) = \mathscr{P}_L$ and $V_R(\mathscr{P}) = \mathscr{P}_R$ through the following procedure:

Initially, let $\mathscr{P} = \mathscr{P}_h$.
While $|\mathscr{P}| < t$ do
  let Q be a path in $\mathscr{P}$ that visits vertices in both V_L and V_R;
  let $Q = Q_1 \dot{+} Q_2$, where one of $end(Q_1)$ and $start(Q_2)$ is in V_L and the other is in V_R;
  let $\mathscr{P} = (\mathscr{P} \backslash \{Q\}) \cup \{Q_1, Q_2\}$;
end While.

It follows immediately from the above procedure that statement (2) holds and the lemma is proved. $\square$

Lemma 3.5. *Assume that G is a distance-hereditary graph formed from G_L and G_R by a true-twin operation and that $\mathscr{P}C_L$ and $\mathscr{P}C_R$ are (t_L, s_L, u_L)-constrained and (t_R, s_R, u_R)-constrained path covers of G_L and G_R, respectively. Then, G has a (t, s, u)-constrained path cover for each t, s, u satisfying the following conditions*:

(1) $\bar{u} = u - u_L - u_R \geqslant 0$;
(2) $s + 2u = (s_L + s_R) + 2(u_L + u_R)$;
(3) *if* $t = s = \bar{u} = 0$, *then* $t_L = t_R = 0$; *otherwise*, $\max\{0, t_L - t_R - s_R + \bar{u}, t_R - t_L - s_L + \bar{u}\} \leqslant t \leqslant t_L + t_R - \max\{0, \bar{u} - s_L, \bar{u} - s_R\}$.

Proof. Let $t, s, u, \bar{u}$ satisfy Conditions (1)–(3). We prove this lemma by showing how to construct a (t, s, u)-constrained path cover of G from $\mathscr{P}C_L$ and $\mathscr{P}C_R$. By Conditions (1) and (2), we obtain that $s = (s_L - \bar{u}) + (s_R - \bar{u})$. Suppose that $t = s = \bar{u} = 0$. Then, $t_L = t_R = 0$. Since $\bar{u} = 0$, $u = u_L + u_R$. Hence, $s = s_L + s_R = 0$. Then, $s_L = s_R = 0$. Clearly, $\mathscr{P}C_L \cup \mathscr{P}C_R$ forms a $(0, 0, u)$-constrained path cover of G, where $u = u_L + u_R$.

In the following, suppose that $t \neq 0$, $s \neq 0$ or $\bar{u} \neq 0$. Then, $\max\{0, t_L - t_R - s_R + \bar{u}, t_R - t_L - s_L + \bar{u}\} \leqslant t \leqslant t_L + t_R - \max\{0, \bar{u} - s_L, \bar{u} - s_R\}$. Hence, we have that

$$t \leqslant t_L + t_R - \max\{0, \bar{u} - s_L, \bar{u} - s_R\}, \tag{9}$$

$$-(t + s_L - \bar{u}) \leqslant t_L - t_R \leqslant t + s_R - \bar{u}. \tag{10}$$

Consider the following three cases:

Case 1: $s_L \geqslant \bar{u}$ and $s_R \geqslant \bar{u}$. By Eq. (9), we get that $t \leqslant t_L + t_R$. Let $S(G_L, \mathscr{P}C_L) = L_s \cup L_{\bar{u}}$ and $S(G_R, \mathscr{P}C_R) = R_s \cup R_{\bar{u}}$ such that $L_s \cap L_{\bar{u}} = \emptyset$, $|L_{\bar{u}}| = \bar{u}$, $R_s \cap R_{\bar{u}} = \emptyset$ and $|R_{\bar{u}}| = \bar{u}$. Then, $|L_s| = s_L - \bar{u}$ and $|R_s| = s_R - \bar{u}$. Let $\mathscr{P}C_{\bar{u}} = L_{\bar{u}} \lhd_{[\bar{u}]} R_{\bar{u}}$ and let $\mathscr{P}C_u = \mathscr{P}C_{\bar{u}} \cup U(G_L, \mathscr{P}C_L) \cup U(G_R, \mathscr{P}C_R)$. Then, $\mathscr{P}C_u$ is a set of vertex-disjoint non-twin paths of G of size u. There are two subcases:

Case 1.1: $t = 0$ and $t_L = t_R$. Suppose that $t_L = 0$. Then, $L_s \cup R_s \cup \mathscr{P}C_u$ forms a $(0, s, u)$-constrained path cover of G. On the other hand, suppose that $t_L > 0$. Let $\mathscr{P}C_h = T(G_L, \mathscr{P}C_L) \bowtie_{[t_L]} T(G_R, \mathscr{P}C_R)$. Then, $\mathscr{P}C_h$ consists of only a twin path P of G with one end vertex in $TS(G_L)$ and the other end vertex in $TS(G_R)$. Let Q be a path in $L_s \cup R_s$ if $|L_s| + |R_s| \neq 0$, i.e., $s \neq 0$; otherwise, let Q be a path in $\mathscr{P}C_{\bar{u}}$. Without loss of generality, let $end(Q)$ be in $TS(G)$ if $Q \in L_s \cup R_s$; otherwise, let $Q = Q_1 \dot{+} Q_2$ so that $end(Q_1)$ is in $TS(G_L)$. By assumption that $t \neq 0$, $s \neq 0$ or $\bar{u} \neq 0$, we have that $s \neq 0$ or $\bar{u} \neq 0$. Hence, Q exists. Let $\mathscr{P}C_s = L_s \cup R_s$. Let$\mathscr{P}C_s = \mathscr{P}C_s \backslash \{Q\} \cup \{Q \dot{+} P\}$ if $Q \in L_s \cup R_s$; otherwise, let $\mathscr{P}C_u = \mathscr{P}C_u \backslash \{Q\} \cup \{Q_1 \dot{+} P \dot{+} Q_2\}$. Then, $\mathscr{P}C_s \cup \mathscr{P}C_u$ forms a $(0, s, u)$-constrained path cover of G. Hence, we can obtain a $(0, s, u)$-constrained path cover of G from $\mathscr{P}C_L$ and $\mathscr{P}C_R$.

Case 1.2: $t \neq 0$ or $t_L \neq t_R$. By Eq. (10), we consider the following three subcases:

Case 1.2.1: $\underline{t \leqslant t_L - t_R \leqslant t + s_R - \bar{u}}$. Since $t \neq 0$ or $t_L \neq t_R$, we have that $t_L \neq t_R$ if $t = 0$. Hence, $t_L - t_R > 0$ in this subcase. Let $\widehat{\mathscr{P}C} = T(G_L, \mathscr{P}C_L) \odot_{[t_R]} T(G_R, \mathscr{P}C_R)$. Then, $\widehat{\mathscr{P}C}$ is a set of vertex-disjoint twin paths of G of size $t_L - t_R$ such that both end vertices of every path in $\widehat{\mathscr{P}C}$ are in $TS(G_L)$. Then, $t \leqslant |\widehat{\mathscr{P}C}| \leqslant t + s_R - \bar{u}$. Let $\widehat{\mathscr{P}C} = \mathscr{P}C_t \cup \mathscr{P}C_{\hat{s}}$ such that $\mathscr{P}C_t \cap \mathscr{P}C_{\hat{s}} = \emptyset$ and $|\mathscr{P}C_t| = t$. Clearly, $0 \leqslant |\mathscr{P}C_{\hat{s}}| \leqslant s_R - \bar{u}$. Let $R_s = R_s \lhd_{[|\mathscr{P}C_{\hat{s}}|]} \mathscr{P}C_{\hat{s}}$ and let $\mathscr{P}C_s = R_s \cup L_s$. Then, $\mathscr{P}C_s$ is a set of vertex-disjoint semi-twin paths of G of size $s = (s_L - \bar{u}) + (s_R - \bar{u})$. Hence, $\mathscr{P}C_t \cup \mathscr{P}C_s \cup \mathscr{P}C_u$ forms a (t, s, u)-constrained path cover of G.

Case 1.2.2: $-t \leqslant t_L - t_R \leqslant t$. If $t = 0$, then $t_L = t_R$ and, hence, it contradicts the assumption that $t_L \neq t_R$. Hence, $t \geqslant 1$. Since $-t \leqslant t_L - t_R \leqslant t$, we have that $t \geqslant t_L - t_R$ and $t \geqslant t_R - t_L$. Hence, $t \geqslant \max\{1, t_L - t_R, t_R - t_L\}$. Since $t \leqslant t_L + t_R$, $\max\{1, t_L - t_R, t_R - t_L\} \leqslant t \leqslant t_L + t_R$. By Lemma 3.4, we can obtain from $T(G_L, \mathscr{P}C_L)$ and $T(G_R, \mathscr{P}C_R)$ a set $\mathscr{P}C_t$ of vertex-disjoint twin paths of G of size t. Let $\mathscr{P}C_s = L_s \cup R_s$. Then, $\mathscr{P}C_t \cup \mathscr{P}C_s \cup \mathscr{P}C_u$ forms a (t, s, u)-constrained path cover of G.

Case 1.2.3: $-(t + s_L - \bar{u}) \leqslant t_L - t_R \leqslant -t$. In this case, $t \leqslant t_R - t_L \leqslant t + s_L - \bar{u}$. By symmetry, we can prove this subcase via arguments similar to those for proving Case 1.2.1.

Case 2: $s_L < \bar{u}$. In this case, $s_R > \bar{u}$ since $s = (s_L - \bar{u}) + (s_R - \bar{u}) \geqslant 0$. By Eq. (9), we get that $t \leqslant t_L + t_R - (\bar{u} - s_L)$. Let $S(G_R, \mathscr{P}C_R) = R_{\bar{u}_1} \cup R_{\bar{u}_2} \cup R_s$ such that $R_{\bar{u}_1} \cap R_{\bar{u}_2} \cap R_s = \emptyset$, $|R_{\bar{u}_1}| = s_L$, $|R_{\bar{u}_2}| = 2(\bar{u} - s_L)$ and $|R_s| = s$. Let $\mathscr{P}C_{\bar{u}_1} = S(G_L, \mathscr{P}C_L) \lhd_{[s_L]} R_{\bar{u}_1}$.Then, $\mathscr{P}C_{\bar{u}_1}$ forms a set of vertex-disjoint non-twin paths of G of size s_L such that one end vertex of every path in $\mathscr{P}C_{\bar{u}_1}$ is in $V_L \backslash TS(G_L)$ and the other end vertex is in $V_R \backslash TS(G_R)$. By Eq. (10), we consider the following two subcases:

Case 2.1: $t+\bar{u}-s_L \leqslant t_L - t_R \leqslant t + s_R - \bar{u} = t + \bar{u} - s_L + s$. In this subcase, $t_L - t_R \geqslant t + \bar{u} - s_L > 0$ since $\bar{u} > s_L$ and $t \geqslant 0$. Let $\widehat{\mathscr{P}C}_t = T(G_L, \mathscr{P}C_L) \odot_{[t_R]} T(G_R, \mathscr{P}C_R)$. Then, $\widehat{\mathscr{P}C}_t$ is a set of vertex-disjoint twin paths of G of size $t_L - t_R$ such that both end vertices of every path in $\widehat{\mathscr{P}C}_t$ are in $TS(G_L)$. Then, $t + \bar{u} - s_L \leqslant |\widehat{\mathscr{P}C}_t| \leqslant t + \bar{u} - s_L + s$. Let $\widehat{\mathscr{P}C}_t = \mathscr{P}C_t \cup L_{\bar{u}} \cup L_s$ such that $\mathscr{P}C_t \cap L_{\bar{u}} \cap L_s = \emptyset$, $|\mathscr{P}C_t| = t$ and $|L_{\bar{u}}| = \bar{u} - s_L$. Then, $0 \leqslant |L_s| \leqslant s$. Let $\mathscr{P}C_s = R_s \lhd_{[|L_s|]} L_s$. Then, $\mathscr{P}C_s$ is a set of vertex-disjoint semi-twin paths of G of size s. Let P be a path of $L_{\bar{u}}$ and let Q_1 and Q_2 be two paths of $R_{\bar{u}_2}$. Then, $Q_1 \dotplus P \dotplus Q_2$ is a non-twin path of G with both end vertices in $V_R \backslash TS(G_R)$. Hence, we can obtain from $L_{\bar{u}}$ and $R_{\bar{u}_2}$ a set $\mathscr{P}C_{\bar{u}_2}$ of vertex-disjoint non-twin paths of G of size $\bar{u} - s_L$ such that both end vertices of every path in $\mathscr{P}C_{\bar{u}_2}$ are in $V_R \backslash TS(G_R)$. Let $\mathscr{P}C_u = \mathscr{P}C_{\bar{u}_1} \cup \mathscr{P}C_{\bar{u}_2} \cup U(G_L, \mathscr{P}C_L) \cup U(G_R, \mathscr{P}C_R)$. Then, $\mathscr{P}C_u$ forms a set of vertex-disjoint non-twin paths of G of size u. Hence, $\mathscr{P}C_t \cup \mathscr{P}C_s \cup \mathscr{P}C_u$ forms a (t, s, u)-constrained path cover of G.

Case 2.2: $-t+\bar{u}-s_L \leqslant t_L - t_R \leqslant t + \bar{u} - s_L$. Suppose $t=0$. Then, $t_L - t_R = \bar{u} - s_L > 0$. Let $\widetilde{\mathscr{P}C}_t = T(G_L, \mathscr{P}C_L) \odot_{[t_R]} T(G_R, \mathscr{P}C_R)$. Then, $\widetilde{\mathscr{P}C}_t$ is a set of vertex-disjoint twin paths of G of size $t_L - t_R = \bar{u} - s_L$ such that both end vertices of every path in $\widetilde{\mathscr{P}C}_t$ are in $TS(G_L)$. Let P be a path of $\widetilde{\mathscr{P}C}_t$ and let Q_1 and Q_2 be two paths of $R_{\bar{u}_2}$. Then, $Q_1 \dotplus P \dotplus Q_2$ is a non-twin path of G with both end vertices in $V_R \backslash TS(G_R)$. Hence, we can obtain from $\widetilde{\mathscr{P}C}_t$ and $R_{\bar{u}_2}$ a set $\mathscr{P}C_{\bar{u}_2}$ of vertex-disjoint non-twin paths of G of size $\bar{u} - s_L$ such that both end vertices of every path in $\mathscr{P}C_{\bar{u}_2}$ are in $V_R \backslash TS(G_R)$. Let $\mathscr{P}C_u = \mathscr{P}C_{\bar{u}_1} \cup \mathscr{P}C_{\bar{u}_2} \cup U(G_L, \mathscr{P}C_L) \cup U(G_R, \mathscr{P}C_R)$. Then, $\mathscr{P}C_u$ forms a set of vertex-disjoint non-twin paths of G of size u. Hence, $R_s \cup \mathscr{P}C_u$ forms a $(0, s, u)$-constrained path cover of G. In the following, suppose that $t > 0$. We first prove that $t_L \geqslant \bar{u} - s_L$. Since $-t + \bar{u} - s_L \leqslant t_L - t_R$, we have that $t_L \geqslant t_R - t + \bar{u} - s_L$. Since $t \leqslant t_L + t_R - (\bar{u} - s_L)$, we get that $t_R - t \geqslant -t_L + \bar{u} - s_L$. Combining the above two equations, we get that $t_L \geqslant -t_L + 2(\bar{u} - s_L)$. Therefore, $t_L \geqslant \bar{u} - s_L$. We next partition $T(G_L, \mathscr{P}C_L)$ into two disjoint subsets, L_t and $L_{\bar{u}}$, such that $|L_{\bar{u}}| = \bar{u} - s_L$. Then, we can obtain a set $\mathscr{P}C_{\bar{u}_2}$ of vertex-disjoint non-twin paths of G of size $\bar{u} - s_L$ by using all paths of $L_{\bar{u}}$ and $R_{\bar{u}_2}$ such that both end vertices of each path in $\mathscr{P}C_{\bar{u}_2}$ are in $V_R \backslash TS(G_R)$. Let $\mathscr{P}C_u = \mathscr{P}C_{\bar{u}_1} \cup \mathscr{P}C_{\bar{u}_2} \cup U(G_L, \mathscr{P}C_L) \cup U(G_R, \mathscr{P}C_R)$. Then, $\mathscr{P}C_u$ is a set of vertex-disjoint non-twin paths of G of size u. Finally, we construct from L_t and $T(G_R, \mathscr{P}C_R)$ a set $\mathscr{P}C_t$ of vertex-disjoint twin paths of G of size t as follows: let $\widehat{t_L} = |L_t|$. Since $t_L = |L_{\bar{u}}| + \widehat{t_L}$ and $|L_{\bar{u}}| = \bar{u} - s_L$, $\widehat{t_L} = t_L - (\bar{u} - s_L)$ and, hence, $\widehat{t_L} - t_R = t_L - t_R - (\bar{u} - s_L)$. Since $-t + \bar{u} - s_L \leqslant t_L - t_R \leqslant t + \bar{u} - s_L$, $-t \leqslant \widehat{t_L} - t_R \leqslant t$. By assumption, $t \geqslant 1$. Hence, $t \geqslant \max\{1, \widehat{t_L} - t_R, t_R - \widehat{t_L}\}$. Since $t \leqslant t_L + t_R - (\bar{u} - s_L)$, $t \leqslant \widehat{t_L} + t_R$. Thus, $\max\{1, \widehat{t_L} - t_R, t_R - \widehat{t_L}\} \leqslant t \leqslant \widehat{t_L} + t_R$. By Lemma 3.4, we can obtain from L_t and $T(G_R, \mathscr{P}C_R)$ a set $\mathscr{P}C_t$ of vertex-disjoint twin paths of G of size t. Then, $\mathscr{P}C_t \cup R_s \cup \mathscr{P}C_u$ forms a (t, s, u)-constrained path cover of G.

Case 3: $s_R < \bar{u}$. This case can be proved by using arguments similar to those used in proving Case 2. □

It follows immediately from Lemmas 3.3 and 3.5 that the following theorem holds:

Theorem 3.6. *Assume G is a distance-hereditary graph formed from G_L and G_R by a true-twin operation. Then, G has a (t, s, u)-constrained path cover if and only if G_L and G_R have (t_L, s_L, u_L)-constrained and (t_R, s_R, u_R)-constrained path covers, respectively, where the following conditions hold*:

(1) $\bar{u} = u - u_L - u_R \geqslant 0$;
(2) $s + 2u = (s_L + s_R) + 2(u_L + u_R)$;
(3) *if* $t = s = \bar{u} = 0$, *then* $t_L = t_R = 0$; *otherwise*, $\max\{0, t_L - t_R - s_R + \bar{u}, t_R - t_L - s_L + \bar{u}\} \leqslant t \leqslant t_L + t_R - \max\{0, \bar{u} - s_L, \bar{u} - s_R\}$.

The above theorem shows the necessary and sufficient condition for G having a constrained path cover while G is formed from G_L and G_R by a true-twin operation. In the following, we will consider the case that G is formed from G_L and G_R by a pendant operation. Note that $TS(G) = TS(G_L)$ if G is formed from G_L and G_R by a pendant operation.

Lemma 3.7. *Assume that G is a distance-hereditary graph formed from G_L and G_R by a pendant operation and that $\mathscr{P}C$ is a (t, s, u)-constrained path cover of G. Then, $V_L(\mathscr{P}C)$ and $V_R(\mathscr{P}C)$ are (t_L, s_L, u_L)-constrained and (t_R, s_R, u_R)-constrained path covers of G_L and G_R, respectively, satisfying the following conditions*:

(1) $\bar{u} = u - u_L - u_R \geqslant 0$;
(2) $t_L \geqslant t$;
(3) $t_L - t_R = t + s_R - \bar{u}$;
(4) $\max\{0, s_L - \bar{u} + s_R - \bar{u}\} \leqslant s \leqslant t_L - t + s_L$ *and* $1 \leqslant s$ *if* $t = \bar{u} = s_L = s_R = 0$, $t_L = t_R$ *and* $t_L \neq 0$.

Proof. By definition, every path in $U(G_L, V_L(\mathscr{PC}))$ and $U(G_R, V_R(\mathscr{PC}))$ only visits vertices in V_L and V_R, respectively, and is a non-twin path of G in $\mathscr{PC}$. Hence, $u \geqslant u_L + u_R$ and Condition (1) is satisfied. Since both end vertices of every path in $T(G, \mathscr{PC})$ are in $TS(G_L)$, every path in $T(G, \mathscr{PC})$ contains at least one twin path of G_L in $V_L(\mathscr{PC})$. Hence, $t_L \geqslant t$ and Condition (2) is satisfied.

We next prove that Condition (3) is satisfied. Let $\bar{U} = U(G, \mathscr{PC}) \backslash (U(G_L, V_L(\mathscr{PC})) \cup U(G_R, V_R(\mathscr{PC})))$. Then, $|\bar{U}| = \bar{u}$. There are six types of paths in $\bar{U}$: type-1 paths are those paths with both end vertices in $V_L \backslash TS(G_L)$; type-2 paths are those paths with one end vertex in $V_L \backslash TS(G_L)$ and the other end vertex in $TS(G_R)$; type-3 paths are those paths with one end vertex in $V_L \backslash TS(G_L)$ and the other end vertex in $V_R \backslash TS(G_R)$; type-4 paths are those paths with both end vertices in $V_R \backslash TS(G_R)$; type-5 paths are those paths with one end vertex in $V_R \backslash TS(G_R)$ and the other end vertex in $TS(G_R)$ and type-6 paths are those paths with both end vertices in $TS(G_R)$. Let u_1, u_2, u_3, u_4, u_5 and u_6 be the numbers of the paths of type-1, type-2, type-3, type-4, type-5 and type-6, respectively. Then, $\bar{u} = u_1 + u_2 + u_3 + u_4 + u_5 + u_6$. There are three types of paths in $S(G, \mathscr{PC})$: type-I paths are those paths with one end vertex in $V_L \backslash TS(G_L)$ and the other end vertex in $TS(G_L)$; type-II paths are those paths with one end vertex in $TS(G_L)$ and the other end vertex in $V_R \backslash TS(G_R)$ and type-III paths are those paths with one end vertex in $TS(G_L)$ and the other end vertex in $TS(G_R)$. Let s_1, s_2 and s_3 be the numbers of the paths of type-I, type-II and type-III, respectively. Then, $s = s_1 + s_2 + s_3$. By the above definitions, we obtain that

$$s_L = s_1 + 2u_1 + u_2 + u_3, \tag{11}$$

$$s_R = s_2 + u_3 + 2u_4 + u_5. \tag{12}$$

Every path of type-3, type-5, type-I and type-III has as many V_L-maximal twin subpaths as V_R-maximal twin subpaths. Every path of type-4 and type-II has exactly one more V_L-maximal twin subpath than V_R-maximal twin subpaths. Every path of type-1, type-2 and type-6 has exactly one more V_R-maximal twin subpath than V_L-maximal twin subpaths. Moreover, every path in $T(G, \mathscr{PC})$ has exactly one more V_L-maximal twin subpath than V_R-maximal twin subpaths since both its end vertices are in $TS(G_L)$. Thus, $t_L - t_R = t + (s_2 + u_4) - (u_1 + u_2 + u_6)$. By Eq. (12), $s_2+u_4=s_R-(u_3+u_4+u_5)$. Since $\bar{u}=u_1+u_2+u_3+u_4+u_5+u_6$, $t_L-t_R=t+(s_R-u_3-u_4-u_5)-(u_1+u_2+u_6)=t+s_R-\bar{u}$ and, hence, Condition (3) is satisfied.

Finally, we prove that Condition (4) is satisfied. By definition, $\bar{u} = u_1 + u_2 + u_3 + u_4 + u_5 + u_6$ and $s = s_1 + s_2 + s_3$. By Eqs. (11) and (12), $s_L - \bar{u} = s_1 + u_1 - u_4 - u_5 - u_6$ and $s_R - \bar{u} = s_2 + u_4 - u_1 - u_2 - u_6$. Hence, $s_L - \bar{u} + s_R - \bar{u} = s_1 + s_2 - u_2 - u_5 - 2u_6 \leqslant s_1 + s_2 \leqslant s$. Clearly, $s \geqslant 0$. Thus, $\max\{0, s_L - \bar{u} + s_R - \bar{u}\} \leqslant s$. By Eq. (11), $s_1 \leqslant s_L$. Since the number of twin paths of G_L in $V_L(T(G, \mathscr{PC}))$ is at least t and every path of type-II and type-III contains at least one twin path of G_L in $V_L(T(G, \mathscr{PC}))$, we have that $s_2 + s_3 \leqslant t_L - t$. Thus, $s = s_1 + s_2 + s_3 \leqslant t_L - t + s_L$. In summary, $\max\{0, s_L - \bar{u} + s_R - \bar{u}\} \leqslant s \leqslant t_L - t + s_L$. On the other hand, suppose that $t = \bar{u} = s_L = s_R = 0$, $t_L = t_R$ and $t_L \neq 0$. By definition and Eqs. (11)–(12), we get that $s_1 = s_2 = 0$ and $u_1 = u_2 = u_3 = u_4 = u_5 = u_6 = 0$. Since $t_L = t_R$, $t_L \neq 0$ and $t = \bar{u} = 0$, we can easily see that $s_3 \geqslant 1$. Hence, $s = s_1 + s_2 + s_3 \geqslant 1$ if $t = \bar{u} = s_L = s_R = 0$, $t_L = t_R$ and $t_L \neq 0$. $\square$

Lemma 3.8. *Assume that G is a distance-hereditary graph formed from G_L and G_R by a pendant operation and that $\mathscr{PC}_L$ and $\mathscr{PC}_R$ are (t_L, s_L, u_L)-constrained and (t_R, s_R, u_R)-constrained path covers of G_L and G_R, respectively. Then, G has a (t, s, u)-constrained path cover for each t, s, u satisfying the following conditions*:

(1) $\bar{u} = u - u_L - u_R \geqslant 0$;
(2) $t_L \geqslant t$;
(3) $t_L - t_R = t + s_R - \bar{u}$;
(4) $\max\{0, s_L - \bar{u} + s_R - \bar{u}\} \leqslant s \leqslant t_L - t + s_L$ *and* $1 \leqslant s$ *if* $t = \bar{u} = s_L = s_R = 0$, $t_L = t_R$ *and* $t_L \neq 0$.

Proof. Let $t, s, u, \bar{u}$ satisfy Conditions (1)–(4). We prove this lemma by showing how to construct a (t, s, u)-constrained path cover of G from $\mathscr{PC}_L$ and $\mathscr{PC}_R$. Suppose that $t = \bar{u} = s_L = s_R = 0$, $t_L = t_R$ and $t_L \neq 0$. By Condition (4), $1 \leqslant s \leqslant t_L$. Let $T(G_L, \mathscr{PC}_L) = L_1 \cup L_2$ and $T(G_R, \mathscr{PC}_R) = R_1 \cup R_2$ such that $L_1 \cap L_2 = \emptyset$, $R_1 \cap R_2 = \emptyset$, $|L_1| = |R_1| = t_L - s + 1$ and $|L_2| = |R_2| = s - 1$. Let $\mathscr{PC}_{s_1} = L_1 \bowtie_{[t_L - s+1]} R_1$, $\mathscr{PC}_{s_2} = L_2 \lhd_{[s-1]} R_2$ and let $\mathscr{PC}_s = \mathscr{PC}_{s_1} \cup \mathscr{PC}_{s_2}$. Then, $\mathscr{PC}_s$ is a set of vertex-disjoint semi-twin paths of G of size s such that one end vertex of every path in $\mathscr{PC}_s$ is in $TS(G_L)$ and the other end vertex is in $TS(G_R)$. Let $\mathscr{PC} = \mathscr{PC}_s \cup U(G_L, \mathscr{PC}_L) \cup U(G_R, \mathscr{PC}_R)$.Then, $\mathscr{PC}$ is a $(0, s, u)$-constrained path cover of G. In the following, we assume that at least one of t, $\bar{u}$, s_L or s_R is not equal to 0 if $t_L = t_R$ and $t_L \neq 0$. Consider the following cases:

Case 1: $s_L \leqslant \bar{u}$ and $s_R \leqslant \bar{u}$. By Conditions (2) and (3), we obtain that $t_L \geqslant t$ and $t_L - t_R = t + s_R - \bar{u}$. Hence, $t_R = t_L - t + \bar{u} - s_R \geqslant \bar{u} - s_R$. By Condition (4), we get that $0 \leqslant s \leqslant t_L - t + s_L$. Let $T(G_L, \mathscr{P}C_L) = \mathscr{P}C_t \cup L_s$ and $T(G_R, \mathscr{P}C_R) = \mathscr{P}C_{\bar{u}_1} \cup R_s$ such that $\mathscr{P}C_t \cap L_s = \emptyset$, $\mathscr{P}C_{\bar{u}_1} \cap R_s = \emptyset$, $|\mathscr{P}C_t| = t$ and $|\mathscr{P}C_{\bar{u}_1}| = \bar{u} - s_R$. Then, $|L_s| = |R_s| = t_L - t = t_R - \bar{u} + s_R$. Since $0 \leqslant s \leqslant t_L - t + s_L$, we consider the following two subcases:

Case 1.1: $s_L < s \leqslant t_L - t + s_L$. Let $L_s = L_{s_1} \cup L_{s_2}$ and $R_s = R_{s_1} \cup R_{s_2}$ such that $L_{s_1} \cap L_{s_2} = \emptyset$, $R_{s_1} \cap R_{s_2} = \emptyset$, $|L_{s_1}| = |R_{s_1}| = t_L - t - s + s_L + 1$ and $|L_{s_2}| = |R_{s_2}| = s - s_L - 1$. Let $\mathscr{P}C_{s_1} = L_{s_1} \bowtie_{[t_L - t - s + s_L + 1]} R_{s_1}$, $\mathscr{P}C_{s_2} = L_{s_2} \lhd_{[s - s_L - 1]} R_{s_2}$, and let $\mathscr{P}C_{\widehat{s}} = \mathscr{P}C_{s_1} \cup \mathscr{P}C_{s_2}$. Then, $\mathscr{P}C_{\widehat{s}}$ is a set of vertex-disjoint semi-twin paths of G of size $s - s_L$ such that one end vertex of every path in $\mathscr{P}C_{\widehat{s}}$ is in $TS(G_L)$ and the other end vertex is in $TS(G_R)$. Let $\mathscr{P}C_s = \mathscr{P}C_{\widehat{s}} \cup S(G_L, \mathscr{P}C_L)$, $\mathscr{P}C_u = \mathscr{P}C_{\bar{u}_1} \cup S(G_R, \mathscr{P}C_R) \cup U(G_L, \mathscr{P}C_L) \cup U(G_R, \mathscr{P}C_R)$ and let $\mathscr{P}C = \mathscr{P}C_t \cup \mathscr{P}C_s \cup \mathscr{P}C_u$. Then, $\mathscr{P}C$ is a (t, s, u)-constrained path cover of G.

*Case*1.2: $0 \leqslant s \leqslant s_L$. Let $\mathscr{P}C_{\widehat{s}} = L_s \bowtie_{[t_L - t]} R_s$ if $t_L - t > 0$; otherwise, let $\mathscr{P}C_{\widehat{s}} = \emptyset$. Then, $\mathscr{P}C_{\widehat{s}}$ consists of only one semi-twin path P of G with one end vertex in $TS(G_L)$ and the other end vertex in $TS(G_R)$ if $\mathscr{P}C_{\widehat{s}} \neq \emptyset$. Let Q be a path in $\mathscr{P}C_t \cup \mathscr{P}C_{\bar{u}_1}$ if $\mathscr{P}C_t \neq \emptyset$ or $\mathscr{P}C_{\bar{u}_1} \neq \emptyset$; otherwise, let Q be a path in $S(G_L, \mathscr{P}C_L) \cup S(G_R, \mathscr{P}C_R)$. We first prove that Q exists if $\mathscr{P}C_{\widehat{s}} \neq \emptyset$. Assume that $\mathscr{P}C_{\widehat{s}} \neq \emptyset$, $\mathscr{P}C_t = \emptyset$ and $\mathscr{P}C_{\bar{u}_1} = \emptyset$. Then, $t_L - t > 0$, $t = 0$ and $\bar{u} - s_R = 0$. By Condition (3), $t_L = t_R$ and $t_L \neq 0$. By assumption, $s_L \neq 0$ or $s_R \neq 0$. Hence, $S(G_L, \mathscr{P}C_L) \cup S(G_R, \mathscr{P}C_R) \neq \emptyset$ and Q exists if $\mathscr{P}C_{\widehat{s}} \neq \emptyset$. Clearly, $end(Q)$ and $start(P)$ are adjacent if P exists. Suppose P exists, i.e., $\mathscr{P}C_{\widehat{s}} \neq \emptyset$. Let $\mathscr{Q}$ be the set of vertex-disjoint paths containing path Q. Then, the positions of both end vertices of path $Q \dotplus P$ in $\mathscr{Q} \backslash \{Q\} \cup \{Q \dotplus P\}$ are the same as those of end vertices of path Q in $\mathscr{Q}$. Note that $\mathscr{Q}$ is either $S(G_L, \mathscr{P}C_L)$, $S(G_R, \mathscr{P}C_R)$, $\mathscr{P}C_t$ or $\mathscr{P}C_{\bar{u}_1}$. Hence, P can be coalesced into one path of $S(G_L, \mathscr{P}C_L) \cup S(G_R, \mathscr{P}C_R) \cup \mathscr{P}C_t \cup \mathscr{P}C_{\bar{u}_1}$ if it exists. Let $\mathscr{Q} = \mathscr{Q} \backslash \{Q\} \cup \{Q \dotplus P\}$ if P exists. We partition $S(G_L, \mathscr{P}C_L)$ into two disjoint sets, $\mathscr{P}C_s$ and $L_{\bar{u}}$, such that $|\mathscr{P}C_s| = s$ and $|L_{\bar{u}}| = s_L - s$. Let $\mathscr{P}C_{\bar{u}} = \mathscr{P}C_{\bar{u}_1} \cup S(G_R, \mathscr{P}C_R)$. Then, $|\mathscr{P}C_{\bar{u}}| = \bar{u}$. By assumption, $s_L \leqslant \bar{u}$ and $0 \leqslant s \leqslant s_L$. Hence, $|L_{\bar{u}}| \leqslant |\mathscr{P}C_{\bar{u}}|$. Let $\mathscr{P}C_{\bar{u}} = \mathscr{P}C_{\bar{u}} \lhd_{[s_L - s]} L_{\bar{u}}$.Then, $\mathscr{P}C_{\bar{u}}$ is a set of vertex-disjoint non-twin paths of G of size $\bar{u}$. Let $\mathscr{P}C_u = \mathscr{P}C_{\bar{u}} \cup U(G_L, \mathscr{P}C_L) \cup U(G_R, \mathscr{P}C_R)$. Then, $\mathscr{P}C_t \cup \mathscr{P}C_s \cup \mathscr{P}C_u$ forms a (t, s, u)-constrained path cover of G.

Case 2: $s_L > \bar{u}$ and $s_R \leqslant \bar{u}$. By Conditions (2) and (3), we obtain that $t_L \geqslant t$ and $t_L - t_R = t + s_R - \bar{u}$. Hence, $t_R \geqslant \bar{u} - s_R$. Let $T(G_L, \mathscr{P}C_L) = \mathscr{P}C_t \cup L_s$ and $T(G_R, \mathscr{P}C_R) = \mathscr{P}C_{\bar{u}_1} \cup R_s$ such that $\mathscr{P}C_t \cap L_s = \emptyset$, $\mathscr{P}C_{\bar{u}_1} \cap R_s = \emptyset$, $|\mathscr{P}C_t| = t$ and $|\mathscr{P}C_{\bar{u}_1}| = \bar{u} - s_R$. Then, $|L_s| = |R_s| = t_L - t = t_R - \bar{u} + s_R$. By Condition (4), we consider the following two subcases:

Case 2.1: $s_L < s \leqslant t_L - t + s_L$. This subcase can be proved by using arguments similar to those used in proving Case 1.1.

Case 2.2: $\max\{0, s_L - \bar{u} + s_R - \bar{u}\} \leqslant s \leqslant s_L$. Let $\mathscr{P}C_{\widehat{s}} = L_s \bowtie_{[t_L - t]} R_s$ if $t_L - t > 0$; otherwise, let $\mathscr{P}C_{\widehat{s}} = \emptyset$. Then, $\mathscr{P}C_{\widehat{s}}$ consists of only one semi-twin path P of G with one end vertex in $TS(G_L)$ and the other end vertex in $TS(G_R)$ if $t_L - t > 0$. Since $s_L - \bar{u} > 0$, $S(G_L, \mathscr{P}C_L) \neq \emptyset$. Let Q be a path in $S(G_L, \mathscr{P}C_L)$ and let $\mathscr{P}C_{s'} = S(G_L, \mathscr{P}C_L) \backslash \{Q\} \cup \{Q \dotplus P\}$. Then, $\mathscr{P}C_{s'}$ is still a set of vertex-disjoint semi-twin paths of G of size s_L such that one end vertex of every path in $\mathscr{P}C_{s'}$ is in $V_L \backslash TS(G_L)$ and the other end vertex is in $TS(G_L)$. We next partition $\mathscr{P}C_{s'}$ into two disjoint subsets, $\mathscr{P}C_s$ and $\bar{U}$, such that $|\mathscr{P}C_s| = s$ and $|\bar{U}| = s_L - s$. We can coalesce two paths in $\bar{U}$ and one path in $\mathscr{P}C_{\bar{u}_1}$ to form a longer non-twin path of G with both end vertices in $V_L \backslash TS(G_L)$. Moreover, we can coalesce one path in $\bar{U}$ and one path in $S(G_R, \mathscr{P}C_R)$ to form a longer non-twin path of G with one end vertex in $V_L \backslash TS(G_L)$ and the other end vertex in $V_R \backslash TS(G_R)$. Since $s_L - \bar{u} + s_R - \bar{u} \leqslant s$, $s_L - s \leqslant 2\bar{u} - s_R = 2(\bar{u} - s_R) + s_R$. Then, all paths of $\bar{U}$ can be coalesced into paths of $\mathscr{P}C_{\bar{u}_1} \cup S(G_R, \mathscr{P}C_R)$. Hence, we can obtain a set $\mathscr{P}C_{\bar{u}}$ of vertex-disjoint non-twin paths of G of size $\bar{u}$ from $\bar{U}$ and $\mathscr{P}C_{\bar{u}_1} \cup S(G_R, \mathscr{P}C_R)$. Let $\mathscr{P}C_u = \mathscr{P}C_{\bar{u}} \cup U(G_L, \mathscr{P}C_L) \cup U(G_R, \mathscr{P}C_R)$. Then, $\mathscr{P}C_t \cup \mathscr{P}C_s \cup \mathscr{P}C_u$ forms a (t, s, u)-constrained path cover of G.

Case 3: $s_L \leqslant \bar{u}$ and $s_R > \bar{u}$. Let $T(G_L, \mathscr{P}C_L) = \mathscr{P}C_t \cup L_{s_1} \cup L_{s_2}$ such that $\mathscr{P}C_t$, L_{s_1} and L_{s_2} are pairwise disjoint, $|\mathscr{P}C_t| = t$ and $|L_{s_1}| = s_R - \bar{u}$. By Condition (3), $|L_{s_2}| = t_L - t - s_R + \bar{u} = t_R$. Let $S(G_R, \mathscr{P}C_R) = \mathscr{P}C_{\bar{u}} \cup R_{s_1}$ such that $\mathscr{P}C_{\bar{u}} \cap R_{s_1} = \emptyset$, $|\mathscr{P}C_{\bar{u}}| = \bar{u}$ and $|R_{s_1}| = s_R - \bar{u}$. Let $\mathscr{P}C_{s_1} = L_{s_1} \lhd_{[s_R - \bar{u}]} R_{s_1}$.Then, $\mathscr{P}C_{s_1}$ is a set of vertex-disjoint semi-twin paths of G of size $s_R - \bar{u}$ such that one end vertex of every path in $\mathscr{P}C_{s_1}$ is in $TS(G_L)$ and the other end vertex is in $V_R \backslash TS(G_R)$. By Condition (4), we consider the following two subcases:

Case 3.1: $s_L + s_R - \bar{u} < s \leqslant t_L - t + s_L$. Let $L_{s_2} = L_a \cup L_b$ and $T(G_R, \mathscr{P}C_R) = R_a \cup R_b$ such that $L_a \cap L_b = \emptyset$, $R_a \cap R_b = \emptyset$, $|L_a| = |R_a| = t_R - s + s_L + s_R - \bar{u} + 1$ and $|L_b| = |R_b| = s - s_L - s_R + \bar{u} - 1$. By Condition (3), $t_R = t_L - t - s_R - \bar{u}$. Hence, $|L_a| = |R_a| = t_L - t + s_L - s + 1$. By assumption of the subcase, $s_L + s_R - \bar{u} + 1 \leqslant s \leqslant t_L - t + s_L$. Hence, $|L_a| = |R_a| \geqslant 1$ and $|L_b| = |R_b| \geqslant 0$. Let $\mathscr{P}C_{s_2} = L_a \bowtie_{[t_R - s + s_L + s_R - \bar{u} + 1]} R_a$,$\mathscr{P}C_{s_3} = L_b \lhd_{[s - s_L - s_R + \bar{u} - 1]} R_b$ and let $\mathscr{P}C_{\widehat{s}} = \mathscr{P}C_{s_2} \cup \mathscr{P}C_{s_3}$.

Then, $\mathscr{PC}_{\widehat{s}}$ is a set of vertex-disjoint semi-twin paths of G of size $s - s_L - s_R + \bar{u}$ such that one end vertex of every path in $\mathscr{PC}_{\widehat{s}}$ is in $TS(G_L)$ and the other end vertex is in $TS(G_R)$. Let $\mathscr{PC}_s = \mathscr{PC}_{s_1} \cup \mathscr{PC}_{\widehat{s}} \cup S(G_L, \mathscr{PC}_L)$. Then, $\mathscr{PC}_s$ is a set of vertex-disjoint semi-twin paths of G of size s. Hence, $\mathscr{PC}_t \cup \mathscr{PC}_s \cup \mathscr{PC}_u$ forms a (t, s, u)-constrained path cover of G.

Case 3.2: $\max\{0, s_L - \bar{u} + s_R - \bar{u}\} \leqslant s \leqslant s_L + s_R - \bar{u}$. Let $\mathscr{PC}_{\widehat{s}} = L_{s_2} \bowtie_{[t_R]} T(G_R, \mathscr{PC}_R)$ if $t_R > 0$; otherwise, let $\mathscr{PC}_{\widehat{s}} = \emptyset$. Then, $\mathscr{PC}_{\widehat{s}}$ consists of only one semi-twin path P of G with one end vertex in $TS(G_L)$ and vertex in $TS(G_R)$ if $t_R > 0$. Since $s_R - \bar{u} > 0$, $\mathscr{PC}_{s_1} \neq \emptyset$. Let Q be a path in $\mathscr{PC}_{s_1}$ and let $\mathscr{PC}_{s_1} = \mathscr{PC}_{s_1} \backslash \{Q\} \cup \{Q \dot{+} P\}$. Then, $\mathscr{PC}_{s_1}$ is a set of vertex-disjoint semi-twin paths of G of size $s_R - \bar{u}$. We next partition $\mathscr{PC}_{s_1} \cup S(G_L, \mathscr{PC}_L)$ into two disjoint subsets, $\mathscr{PC}_s$ and $\bar{U}$, such that $|\mathscr{PC}_s| = s$ and $|\bar{U}| = s_L + s_R - \bar{u} - s$. By assumption of the subcase, $s_L - \bar{u} + s_R - \bar{u} \leqslant s$. Hence, $s_L + s_R - \bar{u} - s \leqslant \bar{u}$. Let $\mathscr{PC}_{\bar{u}} = \mathscr{PC}_{\bar{u}} \lhd_{[|\bar{U}|]} \bar{U}$. Then, $\mathscr{PC}_{\bar{u}}$ is a set of vertex-disjoint non-twin paths of G of size $\bar{u}$. Let $\mathscr{PC}_u = \mathscr{PC}_{\bar{u}} \cup U(G_L, \mathscr{PC}_L) \cup U(G_R, \mathscr{PC}_R)$. Then, $\mathscr{PC}_t \cup \mathscr{PC}_s \cup \mathscr{PC}_u$ forms a (t, s, u)-constrained path cover of G.

Case 4: $s_L > \bar{u}$ and $s_R > \bar{u}$. Let $T(G_L, \mathscr{PC}_L) = \mathscr{PC}_t \cup L_{s_1} \cup L_{s_2}$ such that $\mathscr{PC}_t$, L_{s_1} and L_{s_2} are pairwise disjoint, $|\mathscr{PC}_t| = t$ and $|L_{s_1}| = s_R - \bar{u}$. By Condition (3), $|L_{s_2}| = t_L - t - s_R + \bar{u} = t_R$. Let $S(G_R, \mathscr{PC}_R) = \mathscr{PC}_{\bar{u}} \cup R_{s_1}$ such that $\mathscr{PC}_{\bar{u}} \cap R_{s_1} = \emptyset$, $|\mathscr{PC}_{\bar{u}}| = \bar{u}$ and $|R_{s_1}| = s_R - \bar{u}$. Let $\mathscr{PC}_{s_1} = L_{s_1} \lhd_{[s_R - \bar{u}]} R_{s_1}$.Then, $\mathscr{PC}_{s_1}$ is a set of vertex-disjoint semi-twin paths of G of size $s_R - \bar{u}$ such that one vertex of every path in $\mathscr{PC}_{s_1}$ is in $TS(G_L)$ and the other end vertex is in $V_R \backslash TS(G_R)$. By Condition (4), $s_L - \bar{u} + s_R - \bar{u} \leqslant s \leqslant t_L - t + s_L$. Hence, we consider the following two subcases:

Case 4.1: $s_L + s_R - \bar{u} < s \leqslant t_L - t + s_L$. This subcase can be proved by using arguments similar to those used in proving Case 3.1.

Case 4.2: $s_L - \bar{u} + s_R - \bar{u} \leqslant s \leqslant s_L + s_R - \bar{u}$. Let $\mathscr{PC}_{\widehat{s}} = L_{s_2} \bowtie_{[t_R]} T(G_R, \mathscr{PC}_R)$ if $t_R > 0$; otherwise, let $\mathscr{PC}_{\widehat{s}} = \emptyset$. Then, $\mathscr{PC}_{\widehat{s}}$ consists of only one semi-twin path P of G with one end vertex in $TS(G_L)$ and vertex in $TS(G_R)$ if $t_R > 0$. Since $s_R - \bar{u} > 0$, $\mathscr{PC}_{s_1} \neq \emptyset$. Let Q be a path in $\mathscr{PC}_{s_1}$ and let $\mathscr{PC}_{s_1} = \mathscr{PC}_{s_1} \backslash \{Q\} \cup \{Q \dot{+} P\}$. Then, $\mathscr{PC}_{s_1}$ is a set of vertex-disjoint semi-twin paths of G of size $s_R - \bar{u}$. We next partition $\mathscr{PC}_{s_1} \cup S(G_L, \mathscr{PC}_L)$ into two disjoint subsets, $\mathscr{PC}_s$ and $\bar{U}$, such that $|\mathscr{PC}_s| = s$ and $|\bar{U}| = s_L + s_R - \bar{u} - s$. By assumption, $s_L - \bar{u} + s_R - \bar{u} \leqslant s$. Hence, $s_L + s_R - \bar{u} - s \leqslant \bar{u}$. Let $\mathscr{PC}_{\bar{u}} = \mathscr{PC}_{\bar{u}} \lhd_{[|\bar{U}|]} \bar{U}$. Then, $\mathscr{PC}_{\bar{u}}$ is a set of vertex-disjoint non-twin paths of G of size $\bar{u}$. Let $\mathscr{PC}_u = \mathscr{PC}_{\bar{u}} \cup U(G_L, \mathscr{PC}_L) \cup U(G_R, \mathscr{PC}_R)$. Then, $\mathscr{PC}_t \cup \mathscr{PC}_s \cup \mathscr{PC}_u$ forms a (t, s, u)-constrained path cover of G. $\square$

It follows immediately from Lemmas 3.7 and 3.8 that we get the following theorem:

Theorem 3.9. *Assume G is a distance-hereditary graph formed from G_L and G_R by a pendant operation. Then, G has a (t, s, u)-constrained path cover if and only if G_L and G_R have (t_L, s_L, u_L)-constrained and (t_R, s_R, u_R)-constrained path covers, respectively, where the following conditions hold:*

(1) $\bar{u} = u - u_L - u_R \geqslant 0$;
(2) $t_L \geqslant t$;
(3) $t_L - t_R = t + s_R - \bar{u}$;
(4) $\max\{0, s_L - \bar{u} + s_R - \bar{u}\} \leqslant s \leqslant t_L - t + s_L$ *and* $1 \leqslant s$ *if* $t = \bar{u} = s_L = s_R = 0$, $t_L = t_R$ *and* $t_L \neq 0$.

4. A dynamic programming polynomial-time algorithm

Based on Theorems 3.6 and 3.9, we design an efficient dynamic programming algorithm to compute $\pi(G)$ of a distance-hereditary graph G. By Theorem 2.1, a decomposition tree $DT(G)$ of a distance-hereditary graph G can be constructed in $O(m+n)$ linear time. In the following, assume that the decomposition tree $DT(G)$ of a distance-hereditary graph G is given. For a node v in $DT(G)$, denote by $DT_v(G)$ the subtree of $DT(G)$ rooted at v, and denote by G_v the subgraph of G induced by the leaves of $DT_v(G)$. Our algorithm called PC-DH is sketched as follows: initially, it sets $\mathscr{F}(G_v) = \{(1, 0, 0)\}$ for each leaf v of $DT(G)$. It then visits internal nodes of $DT(G)$ in a postorder sequence. Thus, while visiting a node, both its children were already visited. Suppose that it is about to process internal node v with v_ℓ and v_r being the left and right children of v in $DT(G)$, respectively. Then, it uses $\mathscr{F}(G_{v_\ell})$ and $\mathscr{F}(G_{v_r})$ to compute $\mathscr{F}(G_v)$ by using either Lemma 3.2, 3.5 or 3.8 depending on the label of v. Note that if many pairs of $(t_L, s_L, u_L) \in \mathscr{F}(G_{v_\ell})$ and $(t_R, s_R, u_R) \in \mathscr{F}(G_{v_r})$ produce the same (t, s, u) in $\mathscr{F}(G_v)$, then we remember one and forget the others. If v is the root of $DT(G)$, then it calculates $\pi(G) = \min\{t + s + u | (t, s, u) \in \mathscr{F}(G)\}$ and the algorithm terminates.

The correctness of the above algorithm is based on Theorems 3.6 and 3.9. In the following, we will analyze the complexity of the algorithm.

Lemma 4.1. *Assume G is a distance-hereditary graph formed from G_L and G_R by either a false-twin, a true-twin or a pendant operation. Then, algorithm PC-DH computes $\mathscr{F}(G)$ from $\mathscr{F}(G_L)$ and $\mathscr{F}(G_R)$ in* $\mathrm{O}(n^8)$ *time.*

Proof. Suppose G is formed from G_L and G_R by a false-twin operation. By Lemma 3.2, $\mathscr{F}(G)=\{(t_L+t_R, s_L+s_R, u_L+u_R)|(t_L, s_L, u_L) \in \mathscr{F}(G_L)$ and $(t_R, s_R, u_R) \in \mathscr{F}(G_R)\}$. By definition, $|\mathscr{F}(G_L)| \leqslant \mathrm{O}(n^3)$ and $|\mathscr{F}(G_R)| \leqslant \mathrm{O}(n^3)$. Hence, $\mathscr{F}(G)$ can be computed in $\mathrm{O}(n^6)$ time.

On the other hand, suppose G is formed from G_L and G_R by a true-twin operation. It is obvious from Theorem 3.6 that $\mathscr{F}(G)$ is calculated after processing all pairs of triples in $\mathscr{F}(G_L)$ and $\mathscr{F}(G_R)$. Let $(t_L, s_L, u_L) \in \mathscr{F}(G_L)$ and $(t_R, s_R, u_R) \in \mathscr{F}(G_R)$. Our algorithm will compute all triples (t, s, u) of $\mathscr{F}(G)$ satisfying the conditions given by Lemma 3.5. Obviously, both u and t are integers between 1 and n. For each value of u, s is determined since u_L, u_R, s_L and s_R are fixed. Thus, there are at most $\mathrm{O}(n^2)$ triples (t, s, u) of $\mathscr{F}(G)$ needed to be computed from (t_L, s_L, u_L) and (t_R, s_R, u_R). By definition, $|\mathscr{F}(G_L)| \leqslant n^3$ and $|\mathscr{F}(G_R)| \leqslant n^3$. Hence, our algorithm will refer to the results of Lemma 3.5 at most n^6 times (each of which generates a list of at most n^2 triples), so the list of triples generated for $\mathscr{F}(G)$ is at most n^8. We can use a three-dimensional array to organize the list to eliminate duplicates. Thus, $\mathscr{F}(G)$ can be computed in $\mathrm{O}(n^8)$ time.

The case that G is formed from G_L and G_R by a pendant operation can be proved by using arguments similar to those used in proving the case that G is formed from G_L and G_R by a true-twin operation. □

Since the number of internal nodes in $DT(G)$ is $n-1$ and the time of processing each internal node in $DT(G)$ is bounded in $\mathrm{O}(n^8)$ by Lemma 4.1, $\pi(G)$ can be computed in $\mathrm{O}(n^9)$ time. Though we only describe the algorithm to compute $\pi(G)$, it can be easily extended to find a minimum path cover of a distance-hereditary graph through the constructive proofs of Lemmas 3.5 and 3.8 in the same time bound. Thus, we conclude the following theorem:

Theorem 4.2. *The path cover problem on distance-hereditary graphs can be solved in* $\mathrm{O}(n^9)$ *time.*

Acknowledgment

The authors would like to thank anonymous referee for many useful comments and suggestions which have improved the presentation of this paper.

References

[1] S.R. Arikati, C.P. Rangan, Linear algorithm for optimal path cover problem on interval graphs, Inform. Process. Lett. 35 (1990) 149–153.
[2] H.J. Bandelt, H.M. Mulder, Distance-hereditary graphs, J. Combin. Theory Ser. B 41 (1986) 182–208.
[3] F.T. Boesch, J.F. Gimpel, Covering the points of a digraph with point-disjoint paths and its application to code optimization, J. Assoc. Comput. Mach. 24 (1977) 192–198.
[4] A. Brandstädt, V.B. Le, J.P. Spinrad, Graph Classes: A Survey, SIAM Monographs on Discrete Mathematics and Applications, Philadelphia, 1999.
[5] M.S. Chang, S.Y. Hsieh, G.H. Chen, Dynamic programming on distance-hereditary graphs, in: Lecture Notes in Computer Science, vol. 1350, Springer, Berlin, 1997, pp. 344–353.
[6] D.G. Corneil, H. Lerchs, L.K. Stewart, Complement reducible graphs, Discrete Appl. Math. 3 (1981) 163–174.
[7] D.G. Corneil, Y. Perl, L.K. Stewart, A linear recognition algorithm for cographs, SIAM J. Comput. 14 (1985) 926–934.
[8] B. Courcelle, J. Engelfriet, G. Rozenberg, Handle-rewriting hypergraph grammars, J. Comput. System Sci. 46 (1993) 218–270.
[9] B. Courcelle, J.A. Makowsky, U. Rotics, Linear time solvable optimization problems on graphs of bounded clique-width, Theory Comput. Systems 33 (2000) 125–150.
[10] B. Courcelle, S. Olariu, Upper bounds to the clique width of graphs, Discrete Appl. Math. 101 (2000) 77–114.
[11] P. Damaschke, J.S. Deogun, D. Kratsch, G. Steiner, Finding Hamiltonian paths in cocomparability graphs using the bump number algorithm, Order 8 (1992) 383–391.
[12] G. Damiand, M. Habib, C. Paul, A simple paradigm for graph recognition: Application to cographs and distance-hereditary graphs, Theoret. Comput. Sci. 263 (2001) 99–111.
[13] G. Di Stefano, A routing algorithm for networks based on distance-hereditary topologies, in: Proceedings of the Third International Colloquium on Structural Information and Communication Complexity (SIROCCO'96), Siena, Italy, 1996, pp. 141–151.

[14] A.H. Esfahanian, O.R. Oellermann, Distance-hereditary graphs and multidestination message-routing in multicomputers, J. Combin. Math. Combin. Comput. 13 (1993) 213–222.
[15] W. Espelage, F. Gurski, E. Wanke, How to solve NP-hard graph problems on clique-width bounded graphs in polynomial time, in: Lecture Notes in Computer Science, vol. 2204, Springer, Berlin, 2001, pp. 117–128.
[16] M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, Freeman, San Francisco, CA, 1979.
[17] M.C. Golumbic, Algorithmic Graph Theory and Perfect Graphs, Academic Press, New York, 1980.
[18] M.C. Golumbic, U. Rotics, On the clique-width of some perfect graph classes, Internat. J. Found. Comput. Sci. 11 (2000) 423–443.
[19] S. Goodman, S. Hedetniemi, On the Hamiltonian completion problem, in: Proceedings of the 1973 Capital Conference on Graph Theory and Combinatorics, Lecture Notes in Mathematics, vol. 406, Springer, Berlin, 1974, pp. 262–272.
[20] P.L. Hammer, F. Maffray, Completely separable graphs, Discrete Appl. Math. 27 (1990) 85–99.
[21] E. Howorka, A characterization of Ptolemaic graphs, J. Graph Theory 5 (1981) 323–331.
[22] S.Y. Hsieh, C.W. Ho, T.S. Hsu, M.T. Ko, The Hamiltonian problem on distance-hereditary graphs, Discrete Appl. Math. 154 (2006) 508–524.
[23] R.W. Hung, M.S. Chang, Linear-time algorithms for the Hamiltonian problems on distance-hereditary graphs, Theoret. Comput. Sci. 341 (2005) 411–440.
[24] R.W. Hung, M.S. Chang, Solving the path cover problem on circular-arc graphs by using an approximation algorithm, Discrete Appl. Math. 154 (2006) 76–105.
[25] R.W. Hung, S.C. Wu, M.S. Chang, Hamiltonian cycle problem on distance-hereditary graphs, J. Inform. Sci. Eng. 19 (2003) 827–838.
[26] R. Lin, S. Olariu, G. Pruesse, An optimal path cover algorithm for cographs, Comput. Math. Appl. 30 (1995) 75–83.
[27] S. Moran, Y. Wolfstahl, Optimal covering of cacti by vertex disjoint paths, Theoret. Comput. Sci. 84 (1991) 179–197.
[28] F. Nicolai, Hamiltonian problems on distance-hereditary graphs, Technique Report SM-DU-264, Gerhard-Mercator University, Germany, 1994 (corrected version 1996).
[29] S.C. Ntafos, S. Louis Hakimi, On path cover problems in digraphs and applications to program testing, IEEE Trans. Software Eng. 5 (1979) 520–529.
[30] S.S. Pinter, Y. Wolfstahl, On mapping processes to processors, Internat. J. Parallel Programming 16 (1987) 1–15.
[31] R. Srikant, R. Sundaram, K.S. Singh, C.P. Rangan, Optimal path cover problem on block graphs and bipartite permutation graphs, Theoret. Comput. Sci. 115 (1993) 351–357.
[32] A.S. Tanenbaum, Computer Networks, Prentice-Hall, Englewood Cliffs, NJ, 1981.
[33] P.K. Wong, Optimal path cover problem on block graphs, Theoret. Comput. Sci. 225 (1999) 163–169.
[34] J.H. Yan, G.J. Chang, The path-partition problem in block graphs, Inform. Process. Lett. 52 (1994) 317–322.